Processi di Markov: teoria, analisi e applicazioni

I processi di Markov sono una delle classi più importanti e studiate di processi stocastici nella matematica applicata moderna. La loro denominazione deriva dal matematico russo Andrey Andreyevich Markov (1856-1922), che introdusse questi concetti all'inizio del XX secolo nell'ambito dei suoi studi sulle catene di eventi probabilistici. La proprietà fondamentale che caratterizza questi processi è la cosiddetta "proprietà di Markov" o "assenza di memoria", secondo cui lo stato futuro del sistema dipende unicamente dallo stato presente e non dalla storia passata degli stati attraverso cui il sistema è transitato. La teoria dei processi di Markov ha avuto un impatto rivoluzionario in molteplici discipline scientifiche, dall'ingegneria alla biologia, dall'economia alla fisica, dalla ricerca operativa all'informatica. Questa versatilità deriva dalla capacità di questi modelli matematici di catturare l'essenza di sistemi dinamici soggetti a incertezza, fornendo al contempo strumenti analitici potenti per lo studio del loro comportamento a lungo termine.

Definizione Formale e Proprietà Fondamentali

La Proprietà di Markov

Consideriamo uno spazio di probabilità (Ω, F, P) e uno spazio degli stati S. Un processo stocastico $\{X_t, t \in T\}$ si dice processo di Markov se soddisfa la seguente condizione fondamentale, nota come proprietà di Markov:

$$P(X_{t+s} = j \mid X_t = i, X_{t-1} = i_{t-1}, ..., X_0 = i_0) = P(X_{t+s} = j \mid X_t = i)$$

per ogni scelta di stati $i_0, i_{t-1}, ..., i, j \in S$ e per ogni $s, t \geq 0$. Questa equazione esprime matematicamente il concetto che la distribuzione condizionata dello stato futuro X_{t+s}, dato l'intero passato del processo fino al tempo t, dipende solo dallo stato corrente X_t.

In termini più generali e rigorosi, possiamo definire la proprietà di Markov utilizzando il linguaggio delle σ-algebre. Denotiamo con F_t la σ-algebra generata dalla storia del processo fino al tempo t. La proprietà di Markov può essere espressa come:

$$P(X_{t+s} \in A \mid F_t) = P(X_{t+s} \in A \mid X_t)$$

per ogni insieme misurabile A nella σ-algebra dello spazio degli stati e per ogni $s \geq 0$.

Classificazione dei Processi di Markov

I processi di Markov possono essere classificati secondo diverse caratteristiche fondamentali che ne determinano la struttura matematica e le proprietà analitiche.

Classificazione per Tempo

Quando l'insieme dei parametri temporali T è discreto (tipicamente T = {0, 1, 2, ...} o T = Z), parliamo di catene di Markov a tempo discreto. In questo caso, il processo evolve attraverso transizioni che avvengono a istanti temporali separati e numerabili. La dinamica del sistema è completamente descritta dalle probabilità di transizione ad un passo:

$p_{ij} = P(X_{n+1} = j \mid X_n = i)$

che rappresentano la probabilità di passare dallo stato i allo stato j in un singolo passo temporale.

Quando invece T = [0, ∞) o T = R, abbiamo processi di Markov a tempo continuo. In questo contesto, il sistema può cambiare stato in qualsiasi istante temporale, e la descrizione matematica richiede strumenti più sofisticati come i generatori infinitesimali e le equazioni differenziali stocastiche.

Classificazione per Spazio degli Stati

Lo spazio degli stati S può essere discreto o continuo. Nel cageneraliso discreto, tipicamente S = {1, 2, 3, ...} o un suo sottoinsieme finito, abbiamo le classiche catene di Markov. Questo scenario permette una trattazione algebrica relativamente semplice attraverso matrici di transizione.

Nel caso continuo, dove S potrebbe essere R^n o un suo sottoinsieme, entriamo nel dominio dei processi di diffusione e dei processi di Markov su spazi di Borel. Questi richiedono l'uso del calcolo stocastico e della teoria delle equazioni differenziali stocastiche.

Catene di Markov a Tempo Discreto

Matrice di Transizione e Proprietà Algebriche

Per una catena di Markov a tempo discreto con spazio degli stati finito o numerabile, la dinamica è completamente caratterizzata dalla matrice di transizione $P = (p_{ij})$, dove ogni elemento p_{ij} rappresenta la probabilità di transizione dallo stato i allo stato j in un passo. Questa matrice possiede proprietà fondamentali che derivano dalla natura probabilistica del sistema:

1. Non-negatività: $p_{ij} \geq 0$ per ogni $i, j \in S$
2. Stocasticità per righe: $\Sigma_j p_{ij} = 1$ per ogni $i \in S$

Queste proprietà definiscono quello che in algebra lineare viene chiamato una matrice stocastica. La seconda condizione esprime il fatto che, partendo da qualsiasi stato i, il sistema deve necessariamente transitare in qualche stato (possibilmente rimanendo in i stesso).

Le probabilità di transizione a n passi sono date dalla matrice P^n, ottenuta elevando P alla potenza n-esima. Questo risultato fondamentale, noto come equazione di Chapman-Kolmogorov per catene discrete, stabilisce che:

$p_{ij}^{(n)} = \Sigma_k p_{ik}^{(m)} \cdot p_{kj}^{(n-m)}$

per ogni $0 \leq m \leq n$. Questa equazione esprime il fatto che per andare da i a j in n passi, il processo deve necessariamente passare attraverso qualche stato intermedio k al tempo m.

Classificazione degli Stati

La struttura degli stati di una catena di Markov è fondamentale per comprenderne il comportamento a lungo termine. Introduciamo diverse nozioni di classificazione.

Accessibilità e Comunicazione

Uno stato j è accessibile da uno stato i (scritto $i \to j$) se esiste un intero $n \geq 0$ tale che $p_{ij}^{(n)} > 0$. In altre parole, è possibile raggiungere j partendo da i con probabilità positiva in un numero finito di passi. Due stati i e j comunicano (scritto $i \leftrightarrow j$) se sia $i \to j$ sia $j \to i$. La relazione di comunicazione è una relazione di equivalenza che partiziona lo spazio degli stati in classi di equivalenza chiamate classi comunicanti.

Una catena di Markov si dice irriducibile se tutti gli stati comunicano tra loro, ovvero se esiste una sola classe comunicante che coincide con l'intero spazio degli stati. Questa proprietà è cruciale perché implica che, indipendentemente dallo stato iniziale, il processo può raggiungere qualsiasi altro stato con probabilità positiva.

Periodicità

Il periodo di uno stato i è definito come il massimo comun divisore di tutti gli interi $n \geq 1$ per cui $p_{ii}^{(n)} > 0$. Formalmente:

$$d(i) = \gcd\{n \geq 1 : p_{ii}^{(n)} > 0\}$$

Uno stato si dice aperiodico se $d(i) = 1$, mentre se $d(i) > 1$ lo stato è periodico con periodo $d(i)$. In una catena irriducibile, tutti gli stati hanno lo stesso periodo, che viene chiamato periodo della catena. L'aperiodicità è una condizione tecnica importante per garantire la convergenza delle distribuzioni di probabilità verso una distribuzione limite.

Ricorrenza e Transiency

Uno stato i si dice ricorrente se, partendo da i, il processo ritorna in i con probabilità 1. Matematicamente, definiamo la probabilità di ritorno:

$$f_{ii} = P(T_i < \infty \mid X_0 = i)$$

dove $T_i = \inf\{n \geq 1 : X_n = i\}$ è il tempo del primo ritorno nello stato i. Lo stato i è ricorrente se $f_{ii} = 1$ e transiente se $f_{ii} < 1$.

Un criterio fondamentale per determinare la ricorrenza è basato sul numero atteso di visite allo stato i:

$$N_i = \Sigma_{n=0}^{\infty} p_{ii}^{(n)}$$

Lo stato i è ricorrente se e solo se $N_i = \infty$, mentre è transiente se e solo se $N_i < \infty$. Questo risultato fornisce un metodo pratico per verificare la natura di uno stato attraverso la convergenza o divergenza di una serie.

Ricorrenza Positiva e Nulla

Tra gli stati ricorrenti, distinguiamo ulteriormente tra ricorrenza positiva e ricorrenza nulla. Per uno stato ricorrente i, definiamo il tempo medio di ritorno:

$\mu_i = E[T_i \mid X_0 = i] = \Sigma_{n=1}^{\infty} n \cdot f_{ii}^{(n)}$

dove $f_{ii}^{(n)}$ è la probabilità di ritornare in i per la prima volta esattamente al passo n. Lo stato i si dice ricorrente positivo se $\mu_i < \infty$ e ricorrente nullo se $\mu_i = \infty$.

Gli stati ricorrenti positivi sono fondamentali nello studio del comportamento asintotico delle catene di Markov, poiché solo per questi stati esiste una distribuzione stazionaria ben definita. In una catena finita e irriducibile, tutti gli stati sono necessariamente ricorrenti positivi.

Distribuzioni Stazionarie e Comportamento Limite

Una distribuzione di probabilità $\pi = (\pi_1, \pi_2, ...)$ sullo spazio degli stati S si dice distribuzione stazionaria (o invariante) se soddisfa l'equazione:

$\pi_j = \Sigma_i \pi_i \cdot p_{ij}$ per ogni $j \in S$

In forma matriciale, questa condizione si scrive $\pi = \pi P$, indicando che π è un autovettore sinistro di P corrispondente all'autovalore 1. Se il processo parte con distribuzione iniziale π, allora mantiene questa distribuzione per tutti i tempi successivi, da cui il nome "stazionaria".

Teorema Ergodico Fondamentale

Il teorema ergodico per catene di Markov è uno dei risultati centrali della teoria e stabilisce condizioni sotto cui esiste una distribuzione limite e come questa si relaziona alla distribuzione stazionaria. Il teorema afferma che per una catena di Markov irriducibile, aperiodica e ricorrente positiva, vale:

$\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j$ per ogni $i, j \in S$

dove π è l'unica distribuzione stazionaria della catena, e inoltre $\pi_j = 1/\mu_j$, dove μ_j è il tempo medio di ritorno allo stato j. Questo risultato straordinario afferma che la distribuzione del processo converge alla distribuzione stazionaria indipendentemente dalla distribuzione iniziale.

Inoltre, il teorema ergodico stabilisce una proprietà di media temporale: per qualsiasi funzione f definita sullo spazio degli stati con valore atteso finito rispetto a π, vale la legge dei grandi numeri:

$\lim_{n\to\infty} (1/n) \Sigma_{k=0}^{n-1} f(X_k) = \Sigma_j \pi_j f(j)$ quasi certamente

Questa proprietà permette di interpretare π_j come la proporzione di tempo che il processo trascorre nello stato j nel lungo periodo.

Catene Assorbenti e Tempi di Assorbimento

Una catena di Markov può contenere stati assorbenti, ovvero stati i per cui p_{ii} = 1 e p_{ij} = 0 per ogni j ≠ i. Una volta entrato in uno stato assorbente, il processo vi rimane per sempre. Una catena si dice assorbente se contiene almeno uno stato assorbente e se da ogni stato è possibile raggiungere uno stato assorbente con probabilità positiva.

Per analizzare catene assorbenti, è utile partizionare gli stati in transienti T e assorbenti A. La matrice di transizione può essere scritta in forma canonica:

P = | Q R | | 0 I |

dove Q rappresenta le transizioni tra stati transienti, R le transizioni da stati transienti a stati assorbenti, 0 è una matrice nulla, e I è la matrice identità.

La matrice fondamentale N = (I - Q)^(-1) gioca un ruolo centrale nell'analisi. L'elemento n_{ij} di questa matrice rappresenta il numero atteso di volte che il processo visita lo stato transiente j partendo dallo stato transiente i prima dell'assorbimento. Questa interpretazione deriva dalla serie:

N = I + Q + Q^2 + Q^3 + ... = Σ_{k=0}^∞ Q^k

La matrice B = NR fornisce le probabilità di assorbimento: l'elemento b_{ij} è la probabilità di essere assorbiti nello stato assorbente j partendo dallo stato transiente i.

Il tempo atteso prima dell'assorbimento partendo dallo stato i è dato dalla somma degli elementi della i-esima riga di N:

t_i = Σ_j n_{ij}

Questi risultati hanno applicazioni dirette in problemi di rovina del giocatore, modelli di popolazioni con estinzione, e analisi di affidabilità di sistemi.

Catene di Markov Non-Omogenee

Definizione e Proprietà Fondamentali

Una catena di Markov si dice **non-omogenea** (o non-stazionaria) quando le probabilità di transizione dipendono esplicitamente dal tempo. Formalmente, per una catena a tempo discreto:

P(X_{n+1} = j | X_n = i) = p_{ij}(n)

dove p_{ij}(n) sono le probabilità di transizione al tempo n, che possono variare con n.

La matrice di transizione al tempo n è P(n) = (p_{ij}(n)), e ogni P(n) è una matrice stocastica (elementi non negativi, righe che sommano a 1).

Equazioni di Chapman-Kolmogorov

Per catene non-omogenee, le equazioni di Chapman-Kolmogorov assumono forma più complessa. La probabilità di transizione da i a j in n passi partendo dal tempo m è:

$p_{ij}(m, m+n) = P(X_{m+n} = j \mid X_m = i)$

e soddisfa:

$p_{ij}(m, m+n) = \Sigma_k p_{ik}(m, m+r) \cdot p_{kj}(m+r, m+n)$

per ogni $0 \leq r \leq n$. In forma matriciale:

$P(m, m+n) = P(m, m+r) \cdot P(m+r, m+n)$

Questa generalizzazione riflette il fatto che per calcolare la probabilità di transizione su un intervallo temporale, dobbiamo comporre le transizioni intermedie, ma non possiamo semplicemente elevare a potenza una singola matrice.

Prodotto di Matrici Temporali

Per trovare P(0,n), dobbiamo calcolare il prodotto ordinato:

$P(0,n) = P(n-1) \cdot P(n-2) \cdot ... \cdot P(1) \cdot P(0)$

Nota l'ordine: si applica prima P(0), poi P(1), etc. Questo prodotto generalmente NON commuta, quindi l'ordine è cruciale.

Distribuzione al Tempo n

Se la distribuzione iniziale è $\pi(0) = (\pi_1(0), \pi_2(0), ...)$, la distribuzione al tempo n è:

$\pi(n) = \pi(0) \cdot P(0,n) = \pi(0) \cdot P(n-1) \cdot P(n-2) \cdot ... \cdot P(0)$

Ricorsivamente:

$\pi(n+1) = \pi(n) \cdot P(n)$

Questa equazione mostra che la distribuzione evolve moltiplicando per la matrice di transizione corrente ad ogni passo.

Classificazione degli Stati

La classificazione degli stati per catene non-omogenee è più sottile che per catene omogenee.

Accessibilità Temporale

Uno stato j è **accessibile** da i al tempo (m,n) se p_{ij}(m,n) > 0. Si dice che j è accessibile da i se esiste qualche coppia (m,n) con m < n tale che p_{ij}(m,n) > 0.

La relazione di comunicazione diventa dipendente dal tempo: i e j comunicano se ciascuno è accessibile dall'altro in qualche finestra temporale.

Ricorrenza e Transiency

Per catene non-omogenee, la definizione di ricorrenza considera l'intera sequenza temporale. Sia T_i^{(k)} il k-esimo tempo di ritorno allo stato i. Lo stato i è **ricorrente** se:

P(T_i^{(k)} < ∞ per ogni k | X_0 = i) = 1

e **transiente** altrimenti.

Alternativamente, i è ricorrente se:

Σ_{n=0}^∞ p_{ii}(0,n) = ∞

Comportamento Asintotico

Il comportamento a lungo termine di catene non-omogenee è più complesso e richiede condizioni aggiuntive.

Convergenza Debole

Una condizione sufficiente per convergenza è che le matrici P(n) convergano a una matrice limite P:

lim_{n→∞} P(n) = P

Se P è la matrice di transizione di una catena omogenea ergodica con distribuzione stazionaria π, allora sotto condizioni di regolarità:

lim_{n→∞} π(n) = π

indipendentemente da π(0).

Convergenza Forte

Una condizione più forte è la **convergenza geometrica**: esistono matrice P, vettore π, e costanti C, ρ < 1 tali che:

||P(m,n) - 1π^T|| ≤ C ρ^{n-m}

per ogni m < n, dove 1 è il vettore colonna di tutti 1. Questa garantisce convergenza esponenziale alla distribuzione limite.

Periodicità Asintotica

Catene non-omogenee possono esibire **periodicità asintotica**: le matrici P(n) ciclano attraverso un insieme finito {Q_0, Q_1, ..., Q_{d-1}} con periodo d:

P(n) = Q_{n mod d}

per n sufficientemente grande. In questo caso, la distribuzione π(n) oscilla periodicamente invece di convergere.

Esempi e Applicazioni

Modelli Stagionali

In epidemiologia e ecologia, i tassi di transizione variano stagionalmente. Per esempio, la trasmissione di influenza è maggiore in inverno:

p_{SI}(n) = β(n) S(n)I(n)/N

dove β(n) = β_0[1 + α cos(2πn/52)] modella variazione settimanale con periodo annuale.

Popolazioni con Crescita/Decrescita

Una popolazione con tasso di natalità variabile ha transizioni:

p_{n,n+1}(t) = λ(t) per nascita p_{n,n-1}(t) = μn per morte

dove λ(t) può riflettere disponibilità stagionale di risorse.

Sistemi Economici con Shock

I mercati finanziari sperimentano regimi diversi (boom, recessione, crisi). Una catena per stato economico potrebbe avere:

P(boom, t) durante periodi normali P(crisis, t) durante crisi

con probabilità di transizione che cambiano drasticamente tra regimi.

Code con Arrivi/Servizi Variabili

In sistemi di code, i tassi di arrivo e servizio variano durante il giorno. Un sistema M_t/M_t/1 ha:

Tasso di arrivo: λ(t) Tasso di servizio: μ(t)

dando matrice di transizione tempo-dipendente per il numero di clienti.

Algoritmi di Apprendimento

Algoritmi come simulated annealing usano catene non-omogenee dove la "temperatura" decresce:

$p_{ij}(n) \propto \exp(-[E(j) - E(i)]^+/T(n))$

dove $T(n) \to 0$. Questo permette di sfuggire ottimi locali all'inizio (alta temperatura) e convergere a ottimo globale (bassa temperatura).

Simulated Annealing: Studio Approfondito

Il simulated annealing è forse l'applicazione più importante di catene non-omogenee in ottimizzazione.

Costruzione della Catena

Dato un problema di minimizzazione di $E: S \to R$ su spazio finito S, si costruisce una catena di Markov non-omogenea:

$p_{ij}(n) = q_{ij} \cdot a_{ij}(n)$ per $i \neq j$ $p_{ii}(n) = 1 - \Sigma_{j \neq i} p_{ij}(n)$

dove:

- q_{ij} è probabilità di proposta (simmetrica: $q_{ij} = q_{ji}$)
- $a_{ij}(n) = \min\{1, \exp(-[E(j) - E(i)]/T(n))\}$ è probabilità di accettazione secondo Metropolis
- $T(n)$ è lo schedule di temperatura (temperatura al tempo n)

Teorema di Convergenza di Hajek

Il teorema fondamentale di Hajek (1988) afferma che se lo schedule di temperatura soddisfa:

$T(n) \geq \Delta / \log(n + 2)$

dove Δ è l'altezza della barriera massima che separa un ottimo locale da ottimo globale, allora:

$\lim_{n \to \infty} P(X_n \in S^*) = 1$

dove S^* è l'insieme degli ottimi globali.

La dimostrazione si basa sull'analisi delle transizioni tra bacini di attrazione degli ottimi locali e sul fatto che temperature sufficientemente alte permettono di superare tutte le barriere energetiche con probabilità che decade solo polinomialmente.

Schedule di Temperatura

Logaritmico (teorico): $T(n) = c/\log(n+2)$ garantisce convergenza ma è troppo lento per pratica

Geometrico (pratico): $T(n) = T_0 \alpha^n$ con $0.8 \leq \alpha \leq 0.99$ è molto più veloce ma non ha garanzie teoriche

Adattivo: $T(n+1) = T(n)[1 + T(n)\log(1+\delta)/3\sigma]^{-1}$ dove σ è deviazione standard delle energie recenti

Analisi di Quasi-Stazionarietà

Per temperature fisse $T = T(n) =$ costante, la catena diventa omogenea e ha distribuzione stazionaria di Boltzmann:

$\pi_i(T) = \exp(-E(i)/T) / Z(T)$

dove $Z(T) = \Sigma_j \exp(-E(j)/T)$ è funzione di partizione.

Per $T \to 0$, questa si concentra sugli ottimi globali:

$\lim_{T\to0} \pi_i(T) = 1/|S^*|$ se $i \in S^*$, altrimenti 0

Il simulated annealing "segue" queste distribuzioni quasi-stazionarie mentre T decresce.

Processi di Markov a Tempo Continuo Non-Omogenei

Per processi a tempo continuo, le probabilità di transizione sono funzioni $p_{ij}(s,t)$ dove $s < t$ rappresentano tempo iniziale e finale.

Equazioni di Kolmogorov Tempo-Dipendenti

L'equazione forward è:

$\partial p_{ij}(s,t)/\partial t = \Sigma_k p_{ik}(s,t)\, q_{kj}(t)$

dove $Q(t) = (q_{ij}(t))$ è la matrice di tasso al tempo t.

L'equazione backward è:

$\partial p_{ij}(s,t)/\partial s = -\Sigma_k q_{ik}(s)\, p_{kj}(s,t)$

Soluzione tramite Equazione di Magnus

Per risolvere $P(s,t)$ soggetta a $dP(s,t)/dt = P(s,t)Q(t)$, si usa l'espansione di Magnus:

$P(s,t) = \exp(\Omega(s,t))$

dove:

$\Omega(s,t) = \int_s^t Q(\tau)d\tau + (1/2)\int_s^t \int_s^{\tau_1} [Q(\tau_1), Q(\tau_2)]d\tau_2\, d\tau_1 + ...$

dove $[A,B] = AB - BA$ è il commutatore. Se $[Q(t_1), Q(t_2)] = 0$ per ogni t_1, t_2 (matrici che commutano), allora:

$P(s,t) = \exp(\int_s^t Q(\tau)d\tau)$

Approssimazione Numerica

Per intervalli piccoli δt, l'approssimazione di Euler è:

$P(t, t+\delta t) \approx I + Q(t)\delta t$

e per intervalli finiti:

$P(s,t) \approx \Pi_{k=0}^{n-1} [I + Q(t_k)\delta t]$

dove $t_k = s + k\delta t$ e $n\delta t = t - s$.

Catene Periodicamente Non-Omogenee

Una classe importante è costituita da catene con periodo d:

$P(n) = P(n + d)$ per ogni n

Queste modellano fenomeni ciclici (giorni della settimana, stagioni).

Distribuzione Limite Periodica

Per catene periodicamente non-omogenee ergodiche, esistono d distribuzioni limite $\{\pi^{(0)}, \pi^{(1)}, ..., \pi^{(d-1)}\}$ tali che:

$\lim_{m\to\infty} \pi(md + r) = \pi^{(r)}$ per $r = 0, 1, ..., d-1$

indipendentemente da $\pi(0)$.

Matrice di Transizione Ciclica Media

La matrice prodotto su un ciclo completo:

$\bar{P} = P(d-1)P(d-2)...P(1)P(0)$

è una matrice stocastica che determina le proprietà ergodiche. Se $\bar{P}$ è irriducibile e aperiodica, la catena converge periodicamente.

Metodi Computazionali

Calcolo di Distribuzioni Transitorie

Per calcolare $\pi(n)$, si procede iterativamente:

$\pi(0)$ = distribuzione iniziale

per k da 0 a n-1:

$\pi(k+1) = \pi(k) \cdot P(k)$

Complessità: $O(n \cdot |S|^2)$ dove $|S|$ è numero di stati.

Metodi per Prodotti di Matrici

Per calcolare $P(0,n) = \Pi_{k=0}^{n-1} P(k)$:

Metodo diretto: Moltiplica sequenzialmente, $O(n \cdot |S|^3)$

Divide-and-conquer: Se P(k) hanno struttura particolare (sparse, low-rank), tecniche come Krylov subspace possono ridurre complessità

Approssimazione: Per n grande, approssimare con matrice limite se convergenza è nota

Stima di Parametri

Dati osservazioni $X_0, X_1, ..., X_n$ da una catena non-omogenea, stimare $\{P(k)\}$ è più complesso che per catene omogenee.

Maximum likelihood: Massimizzare:

$L = \Pi_{k=0}^{n-1} p_{X_k, X_{k+1}}(k)$

soggetto a vincoli di stocasticità. Per modelli parametrici $P(k; \theta)$, si massimizza rispetto a θ.

Metodi bayesiani: Usare prior su parametri e aggiornare con osservazioni:

$p(\theta \mid X_0, ..., X_n) \propto L(X_0, ..., X_n \mid \theta)\, p(\theta)$

Simulazione

Per simulare traiettorie da catena non-omogenea:

X_0 = stato iniziale

per n da 0 a N-1:

Genera $U \sim \text{Uniform}(0,1)$

Trova j tale che $\Sigma_{k \le j-1} p_{X_n,k}(n) < U \le \Sigma_{k \le j} p_{X_n,k}(n)$

$X_{n+1} = j$

Nota che la matrice P(n) cambia ad ogni passo.

Applicazioni Avanzate

Controllo Adattivo

In robotica e controllo, l'ambiente cambia e la policy deve adattarsi. Una catena non-omogenea modella:

$X_{n+1} \sim P(X_n, u_n, \theta_n)$

dove θ_n sono parametri stimati adattativamente:

$\theta_{n+1} = \theta_n + \eta_n g(X_n, X_{n+1}, \theta_n)$

combinando esplorazione e sfruttamento.

Filtri Particellari per Sistemi Tempo-Varianti

Per modelli stato-spazio:

$X_{n+1} = f_n(X_n, W_n)$ (dinamica tempo-variante) $Y_n = h_n(X_n, V_n)$ (osservazioni)

il filtro particellare stima $p(X_n \mid Y_1, ..., Y_n)$ usando catena di Markov ausiliaria con transizioni che incorporano osservazioni.

Modelli di Apprendimento Rinforzato Non-Stazionari

In ambienti che evolvono, l'MDP diventa non-stazionario:

$P(s' \mid s, a, t)$ e $R(s, a, t)$

Algoritmi come sliding-window Q-learning o discounted Q-learning adattano la policy:

$Q_{t+1}(s,a) = (1-\alpha_t)Q_t(s,a) + \alpha_t[r_t + \gamma \max_{a'} Q_t(s',a')]$

con learning rate α_t che può variare per tracciare cambiamenti.

Coefficienti di Ergodicità

Il coefficiente di ergodicità di Dobrushin per $P(n)$ è:

$\delta(P(n)) = (1/2) \max_{i,j} \Sigma_k |p_{ik}(n) - p_{jk}(n)|$

misura quanto $P(n)$ "contrae" distribuzioni. Se $\delta(P(n)) < 1$, la matrice è mixing.

Per catene non-omogenee, la contrazione totale su $[0,n]$ è:

$\delta(P(0,n)) \leq \Pi_{k=0}^{n-1} \delta(P(k))$

Se $\Pi_{k=0}^{\infty} \delta(P(k)) = 0$, la catena converge a una distribuzione limite indipendente dallo stato iniziale.

Teorema di Convergenza Generale

Teorema: Sia {X_n} una catena di Markov non-omogenea su spazio finito S con matrici di transizione {P(n)}. Se esistono costanti 0 < ρ < 1 e C > 0 tali che:

δ(P(n)) ≤ ρ per ogni n ≥ N_0

allora esiste una distribuzione limite π tale che:

||π(n) - π||_TV ≤ C ρ^n

per ogni distribuzione iniziale π(0) e n sufficientemente grande.

Condizioni di Mixing Uniforme

Una catena è **uniformly mixing** se esiste funzione ε(k) → 0 tale che per ogni n:

max_{i,j} ||P^{(i)}(n, n+k) - P^{(j)}(n, n+k)||_TV ≤ ε(k)

dove P^{(i)}(n,m) denota distribuzione al tempo m partendo da stato i al tempo n.

Questa proprietà garantisce che l'effetto dello stato iniziale svanisce uniformemente rispetto al tempo di partenza.

Stabilità Geometrica

Una catena è **geometricamente stabile** se esistono funzione di Lyapunov V: S → [0,∞), insieme piccolo C, e costanti λ < 1, b < ∞ tali che:

E[V(X_{n+1}) | X_n = i, n] ≤ λV(i) + b·1_C(i)

per ogni i ∈ S e n ≥ 0.

Sotto questa condizione, la catena visita regolarmente l'insieme C e non può "scappare all'infinito" (per spazi numerabili).

Estensioni e Generalizzazioni

Catene Non-Omogenee con Memoria

Una generalizzazione permette che le transizioni dipendano dalla storia recente:

P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, ..., n) = p_{ij}(n, X_{n-1}, ...)

Questo cattura adattività basata su traiettorie passate, comune in algoritmi di apprendimento.

Esempio: Adaptive MCMC dove la proposta al tempo n dipende dalla varianza empirica delle iterazioni precedenti:

$q_n(x' \mid x) = N(x, \Sigma_n)$

dove $\Sigma_n = \text{Cov}(X_0, ..., X_{n-1})$.

Processi Semi-Markov Non-Omogenei

Combinando non-omogeneità con tempi di permanenza non-esponenziali:

- Catena inclusa: $\{Y_n\}$ con transizioni $P(n)$
- Tempi di permanenza: $T_n \sim F_{Y_n, Y_{n+1}}(\cdot, n)$ dipendenti da tempo

Questi modellano sistemi con sia variabilità temporale che memoria nei tempi di transizione.

Applicazione: Modelli di affidabilità dove i tassi di guasto dipendono dall'età del sistema e le riparazioni richiedono tempi variabili stagionalmente.

Catene Multivariate Non-Omogenee

Per sistemi con multiple componenti interagenti:

$X_n = (X_n^{(1)}, X_n^{(2)}, ..., X_n^{(d)})$

dove ciascuna componente evolve secondo:

$P(X_{n+1}^{(i)} \mid X_n, n) = p_i(X_n, n)$

con accoppiamento attraverso la dipendenza dalle altre componenti.

Esempio: Modelli epidemiologici con multiple malattie che competono per gli stessi host, dove i tassi di trasmissione variano stagionalmente.

Catene con Regime Switching

Una classe importante sono le catene dove le transizioni dipendono da un processo latente (regime):

$P(X_{n+1} \mid X_n, R_n) = P_{R_n}$

dove R_n è una catena di Markov (possibilmente nascosta) che rappresenta il regime corrente.

Esempio: Mercati finanziari con regimi bull/bear dove le transizioni tra stati di prezzo dipendono dal regime macroeconomico.

Confronto: Omogenee vs Non-Omogenee

Proprietà	Catene Omogenee	Catene Non-Omogenee

Matrice di transizione	P costante	P(n) varia con n
Prob. n-passi	P^n	Π_{k=0}^{n-1} P(k)
Dist. stazionaria	πP = π unica	Può non esistere o essere multipla
Convergenza	Automatica per irriducibile, aperiodica	Richiede condizioni aggiuntive
Classificazione stati	Permanente	Può cambiare nel tempo
Complessità calcolo	O(n·	S
Analisi spettrale	Autovalori di P	Complessa, richiede teoria di Floquet
Applicazioni tipiche	Equilibrio, steady-state	Adattamento, ottimizzazione, sistemi evolutivi
Simulazione	Semplice, stessa matrice	Richiede aggiornamento matrice ad ogni passo
Reversibilità	Ben definita	Concetto più complesso

Esempi Dettagliati

Esempio 1: Code con Pattern Giornalieri

Consideriamo un call center con arrivi che variano durante il giorno. Lo stato è il numero di chiamate in coda.

Tassi di arrivo:

- $\lambda(n) = 10$ chiamate/ora per $0 \leq n < 4$ (mattina)
- $\lambda(n) = 30$ chiamate/ora per $4 \leq n < 8$ (mezzogiorno)
- $\lambda(n) = 15$ chiamate/ora per $8 \leq n < 12$ (pomeriggio)
- $\lambda(n) = 5$ chiamate/ora per $n \geq 12$ (sera)

Tasso di servizio: $\mu = 20$ chiamate/ora (costante)

La matrice di transizione al tempo n ha:

p_{i,i+1}(n) = λ(n)/(λ(n) + μ) se i ≥ 1 p_{i,i-1}(n) = μ/(λ(n) + μ) se i ≥ 1 p_{0,1}(n) = λ(n)/(λ(n) + μ) p_{0,0}(n) = μ/(λ(n) + μ)

Questa catena non ha distribuzione stazionaria unica ma cicla giornalmente.

Esempio 2: Genetic Algorithm con Temperatura Decrescente

Un algoritmo genetico per ottimizzazione usa selezione proporzionale alla fitness con intensità variabile:

La probabilità di selezionare individuo i nella generazione n è:

p_i(n) = exp(βf_i/T(n)) / Σ_j exp(βf_j/T(n))

dove f_i è fitness dell'individuo i e T(n) = T_0/(1 + n) è temperatura decrescente.

Per $T(n) \to 0$, la selezione diventa sempre più elitista, convergendo alla selezione deterministica del migliore.

Esempio 3: Modello Epidemiologico Stagionale SIR

Stati: (S, I, R) dove S + I + R = N (popolazione costante)

Transizioni:

- $S \to I$ con probabilità $\beta(n)SI/N$ dove $\beta(n) = \beta_0[1 + \alpha \sin(2\pi n/365)]$
- $I \to R$ con probabilità γ (costante)

La periodicità annuale di β modella stagionalità della trasmissione (più alta in inverno per influenza).

La catena ha d = 365 distribuzioni limite periodiche π^(0), ..., π^(364) corrispondenti ai giorni dell'anno.

Algoritmi Pratici

Algoritmo: Simulated Annealing Completo

Input: Funzione energia $E: S \to R$, temperatura iniziale T_0, schedule α

Output: Stato ottimale approssimato

1. Inizializza $x_0 \in S$ arbitrariamente

2. $T \leftarrow T_0$

3. Per n = 0, 1, 2, ... fino a convergenza:

   a. Genera proposta x' da distribuzione $q(\cdot \mid x_n)$

   b. Calcola $\Delta E = E(x') - E(x_n)$

   c. Se $\Delta E \leq 0$:

      $x_{n+1} \leftarrow x'$ (accetta sempre miglioramenti)

   d. Altrimenti:

      Genera u ~ Uniform(0, 1)

      Se $u < \exp(-\Delta E/T)$:

         $x_{n+1} \leftarrow x'$ (accetta peggioramenti probabilisticamente)

      Altrimenti:

         $x_{n+1} \leftarrow x_n$ (rifiuta)

   e. Aggiorna temperatura: $T \leftarrow \alpha \cdot T$ (schedule geometrico)

   f. Se $|E(x_{n+1}) - E(x_n)| < \varepsilon$ per k iterazioni consecutive:

      STOP (convergenza)

4. Return x_n con $E(x_n)$ minimo tra tutte le iterazioni

Varianti:

- **Parallel tempering**: Mantieni multiple catene a temperature diverse e scambiale
- **Adaptive scheduling**: Aggiusta α basandosi sul tasso di accettazione
- **Restart**: Ricomincia periodicamente da stati casuali per evitare minimi locali

Algoritmo: Stima di Distribuzioni Periodiche

Per catene periodicamente non-omogenee con periodo d:

Input: Sequenza di matrici {P(0), P(1), ..., P(d-1)}, distribuzione iniziale π(0)

Output: Distribuzioni limite periodiche {π^(0), ..., π^(d-1)}

1. Calcola matrice ciclica $\bar{P}$ = P(d-1)·P(d-2)·...·P(0)

2. Trova autovettore sinistro $\bar{\pi}$ di $\bar{P}$ con autovalore 1:

   $\bar{\pi}\cdot\bar{P} = \bar{\pi}$, Σ_i $\bar{\pi}$_i = 1

3. Per r = 0, 1, ..., d-1:

   π^(r) ← $\bar{\pi}$·P(r-1)·P(r-2)·...·P(0) (con indici modulo d)

4. Return {π^(0), ..., π^(d-1)}

Verifica: π^(r)·P(r) = π^((r+1) mod d)

Risultati Teorici Avanzati

Teorema di Lyapunov per Catene Non-Omogenee

Teorema: Sia {X_n} catena non-omogenea su spazio numerabile S. Se esiste funzione V: S → [0, ∞) e costanti λ < 1, b, M < ∞, insieme finito C tali che:

1. E[V(X_{n+1}) | X_n = i, n] ≤ λV(i) + b per ogni i ∉ C, n ≥ 0
2. V(i) ≤ M per ogni i ∈ C

allora la catena è **uniformly geometrically ergodic**: esistono distribuzione limite π e costanti K, ρ < 1 tali che:

||P^{(i)}(n, n+m) - π||_TV ≤ K V(i) ρ^m

per ogni i ∈ S, n, m ≥ 0.

Limite di Diffusione per Catene Non-Omogenee

Sotto rescaling appropriato, catene non-omogenee convergono a processi di diffusione tempo-varianti.

Teorema: Sia {X_n^ε} catena non-omogenea con transizioni:

P(X_{n+1}^ε - X_n^ε = εy | X_n^ε = x, n) = p(x, εn, y)ε + o(ε)

Sotto condizioni di regolarità, il processo continuo X^ε(t) = X_{⌊t/ε⌋}^ε converge in distribuzione a diffusione:

dX(t) = μ(X(t), t)dt + σ(X(t), t)dW(t)

dove:

- $\mu(x,t) = \Sigma_y \, y \cdot p(x,t,y)$
- $\sigma^2(x,t) = \Sigma_y \, y^2 \cdot p(x,t,y)$

Questo giustifica l'uso di approssimazioni diffusive per catene non-omogenee con transizioni piccole e frequenti.

Applicazioni Emergenti

Machine Learning: Non-Stationary Bandits

Nel reinforcement learning con ambienti che cambiano:

Reward al tempo t: $r_t(a) \sim D(\theta_t, a)$

dove θ_t evolve secondo processo (possibilmente non-Markoviano). Algoritmi come:

- **Sliding Window UCB**: Usa solo ultimi W dati
- **Discounted UCB**: Dà peso esponenzialmente decrescente a dati vecchi
- **Change Detection**: Rileva drift nei reward e resetta statistiche

Bioinformatica: Evoluzione con Selezione Variabile

Popolazioni soggette a pressione selettiva variabile nel tempo:

$p_{ij}(t) \propto \exp(s_j(t)/T)$

dove $s_j(t)$ è fitness relativa al tempo t (ambiente variabile).

Modella:

- Resistenza antibiotica con uso intermittente
- Evoluzione virale con stagionalità
- Adattamento a ambienti fluttuanti

Climate Science: Tipping Points in Sistemi Non-Stazionari

Sistemi climatici con forzatura variabile:

$dX_t = f(X_t, \mu(t))dt + \sigma dW_t$

dove $\mu(t)$ rappresenta forcing antropogenico crescente (CO_2, temperatura).

La catena discretizzata è non-omogenea. Punti critici (tipping points) sono transizioni di stato che diventano più probabili al variare di $\mu(t)$.

Conclusioni sulle Catene Non-Omogenee

Le catene di Markov non-omogenee estendono significativamente la teoria classica, permettendo di modellare:

1. **Sistemi evolutivi**: Ambienti che cambiano nel tempo
2. **Algoritmi adattativi**: Parametri che si aggiustano durante esecuzione
3. **Fenomeni stagionali**: Processi con periodicità naturale
4. **Ottimizzazione globale**: Simulated annealing e varianti
5. **Apprendimento**: Sistemi che migliorano con esperienza
6. **Transizioni di fase**: Sistemi che attraversano regimi diversi

Sfide Principali

- **Analisi asintotica**: Più complessa che per catene omogenee
- **Computazione**: Prodotti di matrici invece di potenze
- **Stima**: Più parametri da stimare da dati limitati
- **Teoria spettrale**: Non applicabile direttamente

Direzioni di Ricerca

- Teoria di Floquet per catene periodiche ad alta dimensione
- Metodi di approssimazione per catene con variazione lenta
- Algoritmi efficienti per calcolo di distribuzioni transitorie
- Estensioni a spazi di stati continui e infinito-dimensionali

Processi di Markov a Tempo Continuo

Generatori Infinitesimali e Equazioni di Kolmogorov

Un processo di Markov a tempo continuo $\{X(t), t \geq 0\}$ con spazio degli stati discreto è caratterizzato dalle funzioni di transizione:

$$p_{ij}(t) = P(X(t+s) = j \mid X(s) = i)$$

che, in un processo omogeneo nel tempo, dipendono solo dalla differenza temporale t e non dal punto iniziale s. Queste funzioni soddisfano le equazioni di Chapman-Kolmogorov in forma continua:

$$p_{ij}(t+s) = \Sigma_k \, p_{ik}(t) \cdot p_{kj}(s)$$

che in forma matriciale si scrive $P(t+s) = P(t)P(s)$, indicando che le matrici di transizione formano un semigruppo rispetto alla composizione.

Il generatore infinitesimale Q (o matrice di tasso) è definito attraverso il limite:

$$q_{ij} = \lim_{h \to 0} p_{ij}(h)/h \text{ per } i \neq j$$

$$q_{ii} = \lim_{h \to 0} (p_{ii}(h) - 1)/h$$

Gli elementi fuori diagonale q_{ij} ($i \neq j$) rappresentano i tassi di transizione istantanei dallo stato i allo stato j, mentre gli elementi diagonali $q_{ii} = -\Sigma_{j \neq i} q_{ij}$ sono negativi e rappresentano il tasso totale di uscita dallo stato i.

Le funzioni di transizione soddisfano le equazioni differenziali di Kolmogorov. L'equazione progressiva (o forward) è:

$dP(t)/dt = P(t)Q$

con condizione iniziale $P(0) = I$. L'equazione retrograda (o backward) è:

$dP(t)/dt = QP(t)$

con la stessa condizione iniziale. La soluzione formale di entrambe le equazioni è:

$P(t) = \exp(Qt) = \Sigma_{n=0}^{\infty} (Qt)^n/n!$

dove $\exp(Qt)$ denota l'esponenziale matriciale.

Processi di Nascita e Morte

I processi di nascita e morte costituiscono una classe particolarmente importante di processi di Markov a tempo continuo, dove le transizioni sono possibili solo tra stati adiacenti. Lo spazio degli stati è tipicamente $S = \{0, 1, 2, ...\}$ e le uniche transizioni permesse da uno stato n sono verso n-1 (morte) o n+1 (nascita).

I tassi di nascita λ_n e di morte μ_n caratterizzano completamente il processo. Il generatore infinitesimale ha struttura tridiagonale:

$q_{n,n+1} = \lambda_n$ (tasso di nascita) $q_{n,n-1} = \mu_n$ (tasso di morte) $q_{nn} = -(\lambda_n + \mu_n)$ $q_{ij} = 0$ per $|i-j| > 1$

Le equazioni di bilancio dettagliato per la distribuzione stazionaria π (quando esiste) assumono la forma particolarmente elegante:

$\pi_n \lambda_n = \pi_{n+1} \mu_{n+1}$ per ogni $n \geq 0$

Queste equazioni esprimono il principio di bilancio dei flussi di probabilità: nel lungo periodo, il flusso di probabilità dallo stato n a n+1 deve eguagliare il flusso inverso.

Risolvendo ricorsivamente, otteniamo:

$\pi_n = \pi_0 \cdot (\lambda_0 \lambda_1 ... \lambda_{n-1})/(\mu_1 \mu_2 ... \mu_n)$ per $n \geq 1$

dove π_0 è determinato dalla condizione di normalizzazione $\Sigma_{n=0}^{\infty} \pi_n = 1$. La distribuzione stazionaria esiste se e solo se questa serie converge.

Processo di Poisson

Il processo di Poisson puro di parametro $\lambda > 0$ è un caso speciale di processo di nascita con $\lambda_n = \lambda$ per ogni n e $\mu_n = 0$ (nessuna morte). Il numero di eventi $N(t)$ fino al tempo t segue la distribuzione di Poisson:

$$P(N(t) = n) = (\lambda t)^n e^{-\lambda t} / n!$$

con valore atteso $E[N(t)] = \lambda t$ e varianza $Var[N(t)] = \lambda t$. Gli intervalli tra eventi successivi sono variabili aleatorie indipendenti ed esponenzialmente distribuite con parametro λ.

Il processo di Poisson è il processo di conteggio più importante e trova applicazioni in innumerevoli contesti: arrivi di clienti in sistemi di code, decadimenti radioattivi, chiamate telefoniche, richieste a server web, e molto altro. La proprietà fondamentale è che il processo ha incrementi indipendenti e stazionari, e il numero di eventi in intervalli disgiunti sono variabili aleatorie indipendenti.

Processo di Poisson Composto

Una generalizzazione importante è il processo di Poisson composto, dove ad ogni evento del processo di Poisson è associato un "salto" aleatorio Y_i. Il processo composto è definito come:

$$X(t) = \Sigma_{i=1}^{N(t)} Y_i$$

dove $N(t)$ è un processo di Poisson e $\{Y_i\}$ è una sequenza di variabili aleatorie indipendenti e identicamente distribuite, indipendenti da $N(t)$. Se $E[Y_i] = \mu_Y$ e $Var[Y_i] = \sigma_Y^2$, allora:

$$E[X(t)] = \lambda t \mu_Y \quad Var[X(t)] = \lambda t(\mu_Y^2 + \sigma_Y^2)$$

Questo modello è fondamentale nella teoria del rischio per modellare sinistri assicurativi, in finanza per modellare prezzi di asset con salti, e in operazioni per modellare domanda aggregata.

Tempi di Permanenza e Proprietà Esponenziali

Una proprietà caratteristica dei processi di Markov a tempo continuo è che i tempi di permanenza in ciascuno stato seguono distribuzioni esponenziali. Se il processo entra nello stato i con tasso totale di uscita $q_i = -q_{ii} = \Sigma_{j\neq i} q_{ij}$, allora il tempo T_i che il processo rimane nello stato i prima di transitare altrove è distribuito esponenzialmente con parametro q_i:

$$P(T_i > t) = e^{-q_i t}$$

Questa proprietà è direttamente collegata alla proprietà di assenza di memoria della distribuzione esponenziale: $P(T_i > t+s \mid T_i > t) = P(T_i > s)$. Una volta uscito dallo stato i, il processo salta nello stato j con probabilità:

$$p_{ij}^* = q_{ij}/q_i \text{ per } j \neq i$$

Questo permette di costruire il processo attraverso una catena di Markov inclusa (embedded Markov chain) con matrice di transizione $P^* = (p_{ij}^*)$ e tempi di attesa esponenziali.

Processi di Rinnovo e Teoria dei Semigruppi

I processi di rinnovo generalizzano il processo di Poisson permettendo che gli intervalli tra eventi successivi seguano una distribuzione arbitraria (non necessariamente esponenziale). Sia $\{T_n, n \geq 1\}$ una sequenza di variabili aleatorie indipendenti e identicamente distribuite con funzione di distribuzione F e valore atteso $\mu = E[T_1]$. Il processo di conteggio di rinnovo è definito come:

$$N(t) = \max\{n : T_1 + T_2 + ... + T_n \leq t\}$$

Il teorema fondamentale dei processi di rinnovo, noto come teorema di rinnovo elementare, afferma che:

$$\lim_{t\to\infty} E[N(t)]/t = 1/\mu$$

esprimendo il fatto che il tasso medio di rinnovo nel lungo periodo è l'inverso del tempo medio tra rinnovi.

La funzione di rinnovo $m(t) = E[N(t)]$ soddisfa l'equazione di rinnovo:

$$m(t) = F(t) + \int_0^t m(t-s)\, dF(s)$$

che può essere risolta usando trasformate di Laplace. Il teorema di rinnovo di Blackwell fornisce risultati più forti sulla convergenza della funzione di rinnovo.

Processi di Diffusione e Calcolo Stocastico

Moto Browniano e Processo di Wiener

Il moto Browniano standard $\{W(t), t \geq 0\}$ è il processo di Markov a tempo continuo più importante con spazio degli stati continuo. È definito dalle seguenti proprietà:

1. $W(0) = 0$ quasi certamente
2. Il processo ha incrementi indipendenti: $W(t_4) - W(t_3)$ è indipendente da $W(t_2) - W(t_1)$ per $t_1 < t_2 \leq t_3 < t_4$
3. Gli incrementi sono stazionari: $W(t+s) - W(s)$ ha la stessa distribuzione di $W(t)$
4. $W(t)$ è gaussiano con media 0 e varianza t: $W(t) \sim N(0, t)$
5. Le traiettorie sono continue quasi certamente

Il moto Browniano ha proprietà geometriche notevoli: le traiettorie sono continue ovunque ma differenziabili in nessun punto quasi certamente, hanno variazione quadratica finita uguale a t su $[0,t]$, e variazione totale infinita su qualsiasi intervallo.

Il moto Browniano con drift μ e volatilità σ è definito come:

$$X(t) = \mu t + \sigma W(t)$$

dove μ rappresenta il tasso di crescita deterministico e $\sigma > 0$ controlla l'intensità delle fluttuazioni aleatorie.

Equazioni Differenziali Stocastiche

Le equazioni differenziali stocastiche (SDE) forniscono il framework generale per modellare i processi di diffusione. Una SDE unidimensionale ha la forma:

$$dX(t) = \mu(X(t), t)dt + \sigma(X(t), t)dW(t)$$

dove $\mu(x,t)$ è la funzione di drift, $\sigma(x,t)$ è la funzione di diffusione, e $W(t)$ è un moto Browniano standard. In forma integrale:

$$X(t) = X(0) + \int_0^t \mu(X(s), s)ds + \int_0^t \sigma(X(s), s)dW(s)$$

Il secondo integrale è un integrale stocastico di Itô, definito attraverso un limite in probabilità di somme di Riemann appropriate.

Lemma di Itô

Il lemma di Itô è lo strumento fondamentale del calcolo stocastico, analogo alla regola della catena nel calcolo ordinario. Se $X(t)$ soddisfa la SDE sopra e $f(x,t)$ è una funzione due volte continuamente differenziabile in x e una volta in t, allora il processo $Y(t) = f(X(t), t)$ soddisfa:

$$dY(t) = [\partial f/\partial t + \mu(X,t)\partial f/\partial x + (1/2)\sigma^2(X,t)\partial^2 f/\partial x^2]dt + \sigma(X,t)\partial f/\partial x\, dW(t)$$

$$= \mathrm{Var}_\pi(f) + 2\Sigma_{k=1}^{\infty} \mathrm{Cov}_\pi(f(X_0), f(X_k))$$

dove Var_π e Cov_π denotano varianza e covarianza rispetto alla distribuzione stazionaria. Questo risultato mostra come la dipendenza temporale influenzi la variabilità asintotica.

Il termine $(1/2)\sigma^2(X,t)\partial^2 f/\partial x^2$ è la caratteristica distintiva del calcolo stocastico e deriva dalla variazione quadratica non nulla del moto Browniano: $(dW)^2 = dt$ in senso formale.

Processo di Ornstein-Uhlenbeck

Il processo di Ornstein-Uhlenbeck è un processo di diffusione con ritorno alla media, definito dalla SDE:

$$dX(t) = \theta(\mu - X(t))dt + \sigma dW(t)$$

dove $\theta > 0$ è la velocità di ritorno alla media, μ è il livello di lungo periodo, e $\sigma > 0$ è la volatilità. Questo processo è gaussiano e ha soluzione esplicita:

$$X(t) = X(0)e^{(-\theta t)} + \mu(1 - e^{(-\theta t)}) + \sigma\int_0^t e^{(-\theta(t-s))}dW(s)$$

La distribuzione stazionaria (per $t \to \infty$) è normale con media μ e varianza $\sigma^2/(2\theta)$. Il processo di Ornstein-Uhlenbeck è utilizzato per modellare tassi di interesse, velocità di particelle in fluidi viscosi, e molte altre quantità che tendono a ritornare verso un valore medio.

Equazioni di Kolmogorov per Processi di Diffusione

Per un processo di diffusione generale definito da drift μ(x,t) e coefficiente di diffusione σ(x,t), la densità di transizione p(x,t|y,s) (probabilità di trovare il processo in x al tempo t dato che era in y al tempo s < t) soddisfa due equazioni alle derivate parziali fondamentali.

L'equazione di Fokker-Planck (o equazione di Kolmogorov forward) governa l'evoluzione rispetto alle variabili finali (x,t):

$\partial p/\partial t = -\partial/\partial x[\mu(x,t)p] + (1/2)\partial^2/\partial x^2[\sigma^2(x,t)p]$

L'equazione di Kolmogorov backward governa l'evoluzione rispetto alle variabili iniziali (y,s):

$-\partial p/\partial s = \mu(y,s)\partial p/\partial y + (1/2)\sigma^2(y,s)\partial^2 p/\partial y^2$

Queste equazioni sono fondamentali per calcolare distribuzioni di probabilità, valori attesi, e probabilità di eventi per processi di diffusione, e hanno connessioni profonde con la teoria delle equazioni paraboliche.

Generatori Infinitesimali per Diffusioni

Il generatore infinitesimale L di un processo di diffusione è un operatore differenziale del secondo ordine definito da:

$Lf(x) = \mu(x)f'(x) + (1/2)\sigma^2(x)f''(x)$

Per funzioni f sufficientemente regolari, il generatore determina l'evoluzione temporale del valore atteso:

$d/dt\ E[f(X(t))] = E[Lf(X(t))]$

Il generatore caratterizza completamente il processo di Markov e fornisce uno strumento potente per studiarne le proprietà. In particolare, una funzione armonica per L (tale che Lf = 0) ha la proprietà di martingala: $E[f(X(t)) \mid X(s)] = f(X(s))$ per $t \geq s$.

Leggi dei Grandi Numeri

La legge forte dei grandi numeri per catene di Markov ergodiche afferma che per qualsiasi funzione f con $E_\pi[|f|] < \infty$:

$\lim_{n\to\infty} (1/n) \Sigma_{k=0}^{n-1} f(X_k) = E_\pi[f] = \Sigma_i \pi_i f(i)$ quasi certamente

indipendentemente dalla distribuzione iniziale. Questo risultato giustifica l'uso di medie temporali per stimare valori attesi stazionari, fondamento di metodi Monte Carlo basati su catene di Markov.

Metodi Numerici Avanzati per Processi di Markov

Calcolo di Distribuzioni Stazionarie

Metodi Diretti per Catene Finite

Per catena con matrice P, distribuzione stazionaria π soddisfa:

πP = π, Σ_i π_i = 1

Sistema lineare: Riscrivere come π(P - I) = 0 con vincolo Σπ_i = 1:

[P^T - I | 1]^T [π | λ]^T = [0 | 1]^T

dove λ è moltiplicatore di Lagrange.

Complessità: O(n³) per sistemi densi, O(n) per sistemi sparsi con iterative solvers.

Power Iteration

Iterare π^{(k+1)} = π^{(k)}P fino a convergenza.

Convergenza: Rate geometrico con fattore |λ_2| dove λ_2 è secondo autovalore.

Accelerazione:

- **Aitken's Δ²**: Estrapolare usando tre iterate successive
- **Chebyshev acceleration**: Usa polinomi di Chebyshev per ottimizzare

Tasso accelerato: O(√|λ_2|) invece di O(|λ_2|).

Metodi Iterativi

Gauss-Seidel per Catene Reversibili:

π_i^{(k+1)} = (Σ_{j<i} p_{ji} π_j^{(k+1)} + Σ_{j>i} p_{ji} π_j^{(k)}) / (1 - p_{ii})

converge più velocemente di Jacobi (simultanea).

Successive Over-Relaxation (SOR):

π_i^{(k+1)} = (1-ω)π_i^{(k)} + ω·[Gauss-Seidel update]

con ω ∈ (1,2) ottimizzato per velocizzare convergenza.

Metodi Krylov Subspace

Per sistemi grandi e sparsi, GMRES o BiCGSTAB risolvono:

(P^T - I)π = 0

proiettando su Krylov subspaces K_m = span{r_0, Ar_0, ..., A^{m-1}r_0}.

Vantaggio: Convergenza in O(m) iterazioni con m << n per matrici ben-condizionate.

Decomposizione QR per Generatori

Per processo a tempo continuo con generatore Q, trovare π tale che πQ = 0.

Metodo QR: Fattorizzare Q^T = QR e risolvere triangolare R^Tπ^T = 0 con normalizzazione.

Stabilità numerica: Migliore di metodi diretti su (Q^T - 0I) che è singolare.

Riduzione di Modelli ad Alta Dimensionalità

Lumping (Aggregazione)

Partizionare spazio stati S in aggregati {A_1, ..., A_m} con m << |S|.

Lumping esatto: Se per ogni i,j e aggregato A_k:

Σ_{j'∈A_k} p_{ij'} = costante per ogni i ∈ A_l

allora la catena aggregata è Markoviana con matrice:

P̃_{lk} = Σ_{j∈A_k} p_{ij}/|A_l| per qualsiasi i ∈ A_l

Lumping approssimato: Quando lumping esatto non è possibile, minimizzare errore di approssimazione.

Quasi-Lumpability

La catena è **quasi-lumpable** se:

max_{i,i'∈A_l} |Σ_{j∈A_k} (p_{ij} - p_{i'j})| ≤ ε

per ε piccolo. L'errore di approssimazione è controllato da ε.

Algoritmo:

1. Parti con aggregazione iniziale
2. Raffina iterativamente dividendo aggregati con alta variabilità interna
3. Ferma quando ε < tolleranza

State Space Truncation

Per catene su spazi numerabili infiniti, troncare a {0, 1, ..., N} con condizione riflettente o assorbente a N.

Scelta di N: Basata su probabilità di coda:

π_∞ = Σ_{i>N} π_i < ε

Metodi adaptativi aggiustano N durante calcolo.

Kronecker Product Representation

Per modelli prodotto tensoriale (es: M code indipendenti):

$P = P_1 \otimes P_2 \otimes ... \otimes P_M$

Vantaggio: Storage $O(Mn^2)$ invece di $O(n^M)$, calcolo π con metodi iterativi senza formare P esplicitamente.

Shuffle algorithm: Calcola πP efficientemente sfruttando struttura Kronecker.

Metodi per Processi a Tempo Continuo

Uniformization (Randomization)

Per processo con generatore Q, uniformization trasforma in catena discreta equivalente.

Metodo:

1. Scegli $\lambda \geq \max_i |q_{ii}|$
2. Definisci matrice di transizione $P = I + Q/\lambda$
3. Catena discreta embedded con salti Poisson(λ)

Vantaggio: P è stocastica, permettendo uso di metodi per catene discrete.

Distribuzione al tempo t:

$\pi(t) = \Sigma_{n=0}^{\infty} e^{-\lambda t} (\lambda t)^n/n! \cdot \pi(0)P^n$

Troncando somma: errore controllato da Poisson tail.

Matrix Exponential

Calcolare $P(t) = \exp(Qt)$ per piccoli sistemi.

Metodi:

Padé approximation:

$\exp(Qt) \approx R_p(Qt) = [I - Qt/(2p)]^{-p} [I + Qt/(2p)]^p$

Scaling and squaring:

$\exp(Qt) = [\exp(Qt/2^k)]^{2^k}$

Scegli k tale che $||Qt/2^k|| < 1$, poi usa serie di Taylor per $\exp(Qt/2^k)$.

Krylov subspace methods: Per sistemi sparsi grandi, proiettare su Krylov subspace.

Quasi-Stationary Distributions

Per processi con stati assorbenti, la distribuzione condizionata a non assorbimento converge a **quasi-stationary distribution** π_{QS}:

$\pi_{QS}(j) = \lim_{t\to\infty} P_i(X_t = j \mid \tau > t)$

dove τ è tempo di assorbimento.

Soddisfa:

$\lambda_{QS} \pi_{QS} = \pi_{QS} Q_{TT}$

dove Q_{TT} è sottomatrice di Q ristretta a stati transienti, e λ_{QS} è autovalore dominante.

Applicazioni: Epidemiologia (quasi-estinzione), ecologia (popolazioni persistence).

Metodi Monte Carlo Avanzati

Importance Sampling per Eventi Rari

Per stimare P(A) dove A è evento raro:

Standard MC: Simulare N traiettorie, stimare $P(A) \approx \#(\text{traiettorie in } A)/N$. Alta varianza se $P(A) << 1$.

Importance Sampling:

1. Cambiare misura: simulare sotto Q invece di P
2. Riscalare: $E_P[1_A] = E_Q[1_A \cdot dP/dQ]$

Scelta ottimale: $Q(\cdot) \propto P(\cdot|A)$, ma richiede conoscere P(A).

Change of measure per processi di Markov: Twisting dei tassi di transizione.

Splitting (Multilevel Splitting)

Per stimare P(A) = P(raggiungere livello L partendo da 0):

1. Definire livelli intermedi: $0 < L_1 < L_2 < ... < L_k = L$
2. Stimare $p_i = P(L_i \to L_{i+1})$ per ciascun i
3. $P(A) \approx \Pi_i p_i$

Ad ogni livello, simulare multiple traiettorie (splitting) dalle traiettorie che raggiungono il livello successivo.

Vantaggio: Variance reduction esponenziale per eventi rari.

Rare Event Simulation Algorithm (RESTART)

Combina splitting con threshold adaptation:

threshold ← 0

particles ← {X_0}

While threshold < target:

Simula tutte particles fino a tempo T

Identifica quantile α (es: 0.8) del massimo raggiunto

threshold ← quantile α

Splitting: Duplica particles che hanno superato threshold

Killing: Elimina particles sotto threshold

Estima P(A) come frazione sopravvissuta × adjustment per splitting

Multilevel Monte Carlo (MLMC)

Per stimare E[g(X_T)]:

Idea: Usa simulazioni grossolane (cheap) per varianza reduction di simulazioni fini (expensive).

Decomposizione telescopica:

E[g(X_T)] = E[g(X_T^0)] + Σ_{l=1}^L E[g(X_T^l) - g(X_T^{l-1})]

dove X^l usa timestep Δt_l = T/2^l.

Vantaggio: Complessità O(ε^{-2}) invece di O(ε^{-3}) per accuratezza ε.

Metodi per Catene Grandi e Sparse

Graph Algorithms

Per catene su grafi (random walks, MCMC su combinatorial spaces):

PageRank computation:

π = (1-α)(I - αP)^{-1} v

dove v è personalizzazione vector. Risolvere con:

- Power iteration
- Gauss-Seidel
- Conjugate gradient

Connectivity analysis:

- Tarjan's algorithm per strongly connected components
- Dijkstra per shortest paths (mixing time lower bounds)

Aggregation-Disaggregation

Idea: Alternare tra coarse (aggregato) e fine (originale) livelli.

Inizializza π^0

For k = 0, 1, 2, ...:

 # Aggregation

 π̄ ← aggregate(π^k)

  # Solve coarse problem

 π̄ ← solve(P̄, π̄) # Poche iterazioni

  # Disaggregation

 π^{k+1} ← disaggregate(π̄, π^k)

 If ||π^{k+1} - π^k|| < tol: STOP

Convergenza: O(log ε) livelli di aggregazione, each con costo ridotto.

Adaptive Aggregation

Scegliere aggregazione dinamicamente basandosi su struttura locale:

- Raggruppare stati con transizioni simili
- Usare spectral clustering su grafo delle transizioni
- Aggiustare aggregazione iterativamente

Parallel e GPU Computing

Domain Decomposition

Partizionare spazio stati in domini {D_1, ..., D_p} assegnati a processori diversi.

Schwarz alternating method:

For each processor i in parallel:

Risolvi π_i su D_i con boundary conditions da altri domini

Sincronizza boundary values

Itera fino a convergenza

Overlapping domains: Migliora convergenza a costo di comunicazione extra.

GPU Acceleration

Sparse matrix-vector products: Kernel CUDA per $\pi \leftarrow \pi P$.

Challenges:

- Memory coalescing per accessi efficienti
- Load balancing tra threads
- Communication overhead per matrices grandi

Libraries: cuSPARSE, ViennaCL per operazioni sparse su GPU.

Asynchronous Iteration

Su sistemi distribuiti, permettere processori di aggiornare asincronamente senza sync barriers.

Teorema (Chazan-Miranker): Per matrici con diagonal dominance, async iteration converge sotto condizioni mild.

Vantaggio: Riduce idle time, scala meglio a molti processori.

Error Analysis e Control

A Posteriori Error Estimates

Dopo calcolare π_h (approssimazione), stimare errore $||\pi - \pi_h||$.

Residual-based: Calcolare residuo $r = \pi_h - \pi_h P$ e usare:

$||\pi - \pi_h|| \leq C \, ||r||$

dove C dipende da conditioning di P.

Dual-based: Risolvere problema duale per error representation.

Adaptive Refinement

Basandosi su error estimates, raffinare selettivamente:

State space refinement: Espandere regioni con alto errore locale.

Timestep adaptation: Per processi a tempo continuo, usare timesteps variabili.

Hierarchical basis: Multi-resolution representation con wavelets o adaptive grids.

Best Practices

Numerical Stability

Evitare underflow: Log-space arithmetic per probabilità piccole:

log π_{n+1} = log π_n + log P

Conditioning: Per matrici mal condizionate, usare preconditioning:

M^{-1}(I - P)π = 0

dove M approssima (I - P) ma è easy to invert.

Convergence Diagnostics

Per power iteration: Monitorare ||π^{k+1} - π^k||_1 e ||π^k P - π^k||_1.

Per MCMC:

- Trace plots
- Gelman-Rubin statistic (multiple chains)
- Effective sample size
- Autocorrelation plots

Verification

Method of manufactured solutions: Costruire problema con soluzione nota, verificare convergenza rate.

Comparison con metodi indipendenti: Power iteration vs Krylov vs direct solve.

Physical constraints: π ≥ 0, Σπ = 1, πP = π.

Case Studies

Esempio 1: Protein Folding (10^6 stati)

Setup: Markov State Model da simulazioni MD, ~10^6 microstati.

Approccio:

1. Lumping a ~1000 macrostati usando PCCA+
2. Calcolo π con power iteration (converge in ~100 iterazioni)
3. Analisi spettrale per timescales di folding

Challenge: Identificare aggregazione che preserva dinamica lenta.

Esempio 2: PageRank (10^10 nodi)

Setup: Web graph con billions di pages.

Approccio:

1. Sparse storage (CSR format)
2. Power iteration parallelizzato su cluster
3. Dangling nodes handling con personalization vector
4. Convergenza in ~50 iterazioni

Optimization: Block updates, preconditioning con previous PageRank.

Esempio 3: MCMC Diagnostics

Setup: Bayesian posterior sampling da modello complesso.

Tools:

- ESS (effective sample size) da autocorrelazione
- Split-R̂ per within/between chain variance
- Trace plots per visual inspection
- Geweke test per stationarity

Action: Se diagnostics falliscono, increase burn-in, thin chain, o cambiare sampler.

Conclusioni sui Metodi Numerici

I metodi numerici moderni permettono di:

- Risolvere modelli con milioni di stati
- Sfruttare struttura (sparsità, simmetria, Kronecker)
- Parallelizzare efficacemente
- Quantificare accuratezza e controllare errori

Le sfide principali:

- Curse of dimensionality per sistemi veramente grandi
- Identificare buone approssimazioni low-rank
- Balance tra accuratezza e costo computazionale
- Validazione di risultati numerici

Direzioni future:

- Machine learning per model reduction
- Quantum algorithms per linear systems
- Exascale computing per real-time applications
- Automatic differentiation per sensitivity analysis

Applicazioni in Teoria delle Code

Sistemi M/M/1 e Formule di Little

La teoria delle code studia sistemi in cui entità (clienti, pacchetti, richieste) arrivano per ricevere servizio da una o più stazioni. I processi di Markov forniscono il framework matematico naturale per analizzare questi sistemi.

Il sistema M/M/1 è il modello di coda più fondamentale: arrivi secondo un processo di Poisson con tasso λ, servizio esponenziale con tasso μ, un singolo server. Il numero di clienti nel sistema X(t) è un processo di nascita-morte con:

λ_n = λ per ogni n ≥ 0 μ_n = μ per ogni n ≥ 1 μ_0 = 0

La condizione di stabilità è ρ = λ/μ < 1, dove ρ è l'intensità di traffico. Sotto questa condizione, esiste una distribuzione stazionaria geometrica:

π_n = (1-ρ)ρ^n per n = 0, 1, 2, ...

Le prestazioni del sistema sono caratterizzate da:

Numero medio di clienti nel sistema: L = E[X] = ρ/(1-ρ) = λ/(μ-λ) Numero medio in coda: L_q = ρ^2/(1-ρ) = λ^2/(μ(μ-λ)) Tempo medio nel sistema: W = 1/(μ-λ) Tempo medio in coda: W_q = λ/(μ(μ-λ))

Queste quantità sono legate dalle formule di Little, relazioni fondamentali valide per sistemi di coda molto generali:

L = λW L_q = λW_q

che esprimono il principio che il numero medio di entità nel sistema è uguale al tasso di arrivo moltiplicato per il tempo medio di permanenza.

Reti di Code e Teorema di Jackson

Le reti di code estendono l'analisi a sistemi con multiple stazioni di servizio interconnesse. Una rete di Jackson è composta da m stazioni, dove la stazione i ha c_i server identici e tempo di servizio esponenziale con tasso μ_i. I clienti arrivano dall'esterno alla stazione i secondo un processo di Poisson con tasso r_i, e dopo il servizio alla stazione i vanno alla stazione j con probabilità p_{ij} o lasciano la rete con probabilità p_{i0} = 1 - Σ_j p_{ij}.

Il tasso totale di arrivo alla stazione i (esterno + interno) è λ_i, determinato dalle equazioni di bilancio del traffico:

λ_i = r_i + Σ_j λ_j p_{ji} per i = 1, ..., m

Il teorema di Jackson afferma che se ρ_i = λ_i/(c_i μ_i) < 1 per ogni stazione, la distribuzione stazionaria del vettore di stato (X_1, ..., X_m) ha forma prodotto:

π(n_1, ..., n_m) = π_1(n_1) · π_2(n_2) · ... · π_m(n_m)

dove π_i(n_i) è la distribuzione stazionaria della stazione i considerata come sistema isolato. Questo risultato notevole implica che, in equilibrio, le stazioni si comportano come se fossero indipendenti, nonostante la chiara dipendenza nella dinamica.

Sistemi M/G/1 e Formula di Pollaczek-Khinchin

Il sistema M/G/1 generalizza M/M/1 permettendo una distribuzione di servizio generale (non necessariamente esponenziale). Gli arrivi sono Poissoniani con tasso λ, e il tempo di servizio ha media E[S] = 1/μ e secondo momento E[S^2].

La distribuzione stazionaria del numero di clienti nel sistema non ha forma chiusa generale, ma la funzione generatrice delle probabilità e i momenti possono essere calcolati. Il numero medio di clienti nel sistema è dato dalla formula di Pollaczek-Khinchin:

L = ρ + (λ^2 E[S^2])/(2(1-ρ))

dove ρ = λ/μ. Questa formula mostra come la variabilità del servizio (attraverso E[S^2]) influenzi le prestazioni: maggiore variabilità implica code più lunghe. Per servizio esponenziale, E[S^2] = 2/μ^2, e ritroviamo la formula M/M/1.

Applicazioni in Finanza Matematica

Modelli di Pricing di Opzioni

I processi di Markov sono fondamentali nella finanza quantitativa per modellare l'evoluzione dei prezzi di asset e valutare strumenti derivati.

Modello di Black-Scholes-Merton

Il modello più celebre assume che il prezzo di un'azione S(t) segua un moto browniano geometrico:

dS(t) = μS(t)dt + σS(t)dW(t)

dove μ è il drift (rendimento atteso) e σ è la volatilità. Applicando il lemma di Itô a log(S(t)), otteniamo:

log(S(t)) = log(S(0)) + (μ - σ^2/2)t + σW(t)

quindi S(t) = S(0)exp((μ - σ^2/2)t + σW(t)), e il prezzo futuro è lognormalmente distribuito.

La formula di Black-Scholes per il prezzo di un'opzione call europea con strike K e scadenza T è:

C(S,t) = SΦ(d_1) - Ke^(-r(T-t))Φ(d_2)

dove: d_1 = [log(S/K) + (r + σ^2/2)(T-t)] / (σ√(T-t)) d_2 = d_1 - σ√(T-t) Φ è la funzione di distribuzione cumulativa normale standard r è il tasso di interesse privo di rischio

Questa formula è derivata risolvendo l'equazione alle derivate parziali di Black-Scholes, che è essenzialmente l'equazione di Kolmogorov backward per il processo S(t) sotto la misura risk-neutral.

Modelli con Salti

Il modello di Merton estende Black-Scholes incorporando salti nel prezzo tramite un processo di Poisson composto:

dS(t) = μS(t)dt + σS(t)dW(t) + S(t-)dJ(t)

dove J(t) = Σ_{i=1}^{N(t)} (Y_i - 1) e Y_i rappresentano i moltiplicatori del salto (log(Y_i) tipicamente normale). Questo modello cattura meglio gli eventi estremi osservati nei mercati finanziari.

Modelli di Struttura a Termine dei Tassi di Interesse

Modello di Vasicek

Il modello di Vasicek assume che il tasso di interesse a breve termine r(t) segua un processo di Ornstein-Uhlenbeck:

dr(t) = a(b - r(t))dt + σdW(t)

dove a è la velocità di ritorno alla media, b è il livello di lungo periodo, e σ è la volatilità. I prezzi dei bond zero-coupon hanno soluzione analitica:

P(t,T) = A(t,T)e^(-B(t,T)r(t))

dove B(t,T) e A(t,T) sono funzioni deterministiche di t e T determinate dall'equazione di Kolmogorov.

Modello di Cox-Ingersoll-Ross (CIR)

Il modello CIR modifica Vasicek usando una volatilità proporzionale alla radice del tasso:

dr(t) = a(b - r(t))dt + σ√(r(t))dW(t)

Questa specifica garantisce che r(t) rimanga positivo se 2ab ≥ σ^2 (condizione di Feller). Il processo r(t) ha distribuzione stazionaria gamma, e i prezzi dei bond hanno ancora forma esponenziale affine.

Processi di Lévy in Finanza

I processi di Lévy generalizzano il moto browniano permettendo discontinuità (salti). Un processo di Lévy X(t) ha incrementi indipendenti e stazionari, inizia da zero, ed è stocasticamente continuo. La decomposizione di Lévy-Itô rappresenta qualsiasi processo di Lévy come somma di tre componenti:

$X(t) = at + \sigma W(t) + \int_{|x|<1} x(N(t,dx) - tv(dx)) + \int_{|x|\geq 1} xN(t,dx)$

dove a è il drift, σW(t) è una componente browniana, N è una misura aleatoria di Poisson con misura di Lévy v, rappresentante i salti. I primi due termini catturano salti piccoli centrati, il terzo i salti grandi.

Il modello di Variance Gamma, il modello Normal Inverse Gaussian (NIG), e il modello di Merton sono esempi di modelli basati su processi di Lévy usati per catturare le "code grasse" e l'asimmetria osservate nelle distribuzioni dei rendimenti.

Applicazioni in Apprendimento Automatico e Intelligenza Artificiale

Modelli di Markov Nascosti (HMM)

Un modello di Markov nascosto è un processo stocastico bivariate $\{(X_n, Y_n), n \geq 0\}$ dove:

- X_n è una catena di Markov non osservabile (stati nascosti) con matrice di transizione P e distribuzione iniziale π
- Y_n sono osservazioni che dipendono solo da X_n attraverso la matrice di emissione B, dove $b_{ij} = P(Y_n = j \mid X_n = i)$

Gli HMM sono fondamentali per:

- Riconoscimento vocale: gli stati nascosti rappresentano fonemi, le osservazioni sono caratteristiche acustiche
- Bioinformatica: identificazione di regioni geniche, predizione di strutture proteiche
- Finanza: identificazione di regimi di mercato (bull/bear)
- Natural Language Processing: part-of-speech tagging, entity recognition

Algoritmo Forward-Backward

Dato un HMM con parametri $\lambda = (\pi, P, B)$ e una sequenza di osservazioni $Y_1, ..., Y_T$, l'algoritmo forward calcola ricorsivamente:

$\alpha_t(i) = P(Y_1, ..., Y_t, X_t = i \mid \lambda)$

attraverso: $\alpha_1(i) = \pi_i b_{i,Y_1}$ $\alpha_{t+1}(j) = [\Sigma_i \alpha_t(i)p_{ij}]b_{j,Y_{t+1}}$

L'algoritmo backward calcola:

$\beta_t(i) = P(Y_{t+1}, ..., Y_T \mid X_t = i, \lambda)$

attraverso: $\beta_T(i) = 1$ $\beta_t(i) = \Sigma_j p_{ij}b_{j,Y_{t+1}}\beta_{t+1}(j)$

La probabilità totale è $P(Y_1, ..., Y_T \mid \lambda) = \Sigma_i \alpha_T(i)$.

Algoritmo di Viterbi

L'algoritmo di Viterbi trova la sequenza di stati più probabile $X_1^, ..., X_T^$ data la sequenza osservata, risolvendo:

$\max_{X_1,...,X_T} P(X_1, ..., X_T \mid Y_1, ..., Y_T, \lambda)$

Usa programmazione dinamica:

$\delta_t(i) = \max_{X_1,...,X_{t-1}} P(X_1, ..., X_{t-1}, X_t = i, Y_1, ..., Y_t \mid \lambda)$

con ricorsione: $\delta_{t+1}(j) = [\max_i \delta_t(i)p_{ij}]b_{j,Y_{t+1}}$

Mantenendo i "puntatori" $\psi_{t+1}(j) = \text{argmax}_i \delta_t(i)p_{ij}$, si ricostruisce la sequenza ottimale con backtracking.

Algoritmo Baum-Welch (EM per HMM)

L'algoritmo Baum-Welch stima i parametri $\lambda = (\pi, P, B)$ da dati osservati usando il principio di massima verosimiglianza. È un'istanza dell'algoritmo Expectation-Maximization:

E-step: Calcola le probabilità a posteriori degli stati usando forward-backward $\gamma_t(i) = P(X_t = i \mid Y_1, ..., Y_T, \lambda) = \alpha_t(i)\beta_t(i) / P(Y_1, ..., Y_T \mid \lambda)$ $\xi_t(i,j) = P(X_t = i, X_{t+1} = j \mid Y_1, ..., Y_T, \lambda)$

M-step: Aggiorna i parametri $\hat{\pi}_i = \gamma_1(i)$ $\hat{p}_{ij} = \Sigma_{t=1}^{T-1} \xi_t(i,j) / \Sigma_{t=1}^{T-1} \gamma_t(i)$ $\hat{b}_{ij} = \Sigma_{t:Y_t=j} \gamma_t(i) / \Sigma_t \gamma_t(i)$

Processi di Decisione di Markov (MDP)

Un MDP è una quintupla (S, A, P, R, γ) dove:

- S è lo spazio degli stati
- A è lo spazio delle azioni
- P(s'|s,a) è la probabilità di transizione allo stato s' dato stato s e azione a
- R(s,a,s') è la ricompensa immediata
- $\gamma \in [0,1)$ è il fattore di sconto

L'obiettivo è trovare una policy $\pi: S \rightarrow A$ che massimizza il valore atteso del ritorno scontato:

$V^\pi(s) = E[\Sigma_{t=0}^\infty \gamma^t R(S_t, \pi(S_t), S_{t+1}) \mid S_0 = s]$

Equazioni di Bellman

La funzione valore ottimale V*(s) soddisfa l'equazione di Bellman:

$V^*(s) = \max_a \Sigma_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^*(s')]$

La funzione Q ottimale Q*(s,a) rappresenta il valore di prendere l'azione a nello stato s e poi seguire la policy ottimale:

$Q^*(s,a) = \Sigma_{s'} P(s'|s,a)[R(s,a,s') + \gamma \max_{a'} Q^*(s',a')]$

La policy ottimale è deterministica: $\pi^*(s) = \text{argmax}_a Q^*(s,a)$.

Value Iteration e Policy Iteration

Value iteration risolve l'equazione di Bellman iterativamente:

$V_{k+1}(s) = \max_a \Sigma_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_k(s')]$

iniziando da V_0 arbitrario. Sotto condizioni standard, V_k converge a V^*.

Policy iteration alterna tra:

1. Policy evaluation: risolvere V^{π} per la policy corrente π
2. Policy improvement: $\pi'(s) = \text{argmax}_a \Sigma_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^{\pi}(s')]$

Entrambi gli algoritmi convergono alla policy ottimale in spazi di stati finiti.

Q-Learning e Reinforcement Learning

In molte applicazioni, il modello (P e R) non è noto. Il Q-learning è un algoritmo model-free che apprende Q* direttamente dall'esperienza. L'update è:

$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$

dove α è il learning rate, s' è lo stato successivo, e r è la ricompensa osservata. Con appropriate condizioni su α e sufficiente esplorazione, Q converge a Q* quasi certamente.

Deep Q-Networks (DQN) approssimano Q con reti neurali profonde, permettendo di scalare a spazi di stati enormi (come immagini in videogiochi). Algoritmi più recenti come Proximal Policy Optimization (PPO) e Soft Actor-Critic (SAC) hanno ulteriormente avanzato il campo del reinforcement learning profondo.

Catene di Markov Monte Carlo (MCMC)

Le catene di Markov Monte Carlo sono metodi computazionali per campionare da distribuzioni di probabilità complesse, fondamentali in inferenza bayesiana e fisica statistica.

Algoritmo di Metropolis-Hastings

Dato una distribuzione target $\pi(x)$ (nota a meno di costanti di normalizzazione) e una distribuzione di proposta $q(x'|x)$, l'algoritmo genera una catena di Markov con distribuzione stazionaria π:

1. Proponi x' ~ q(x'|x_t)
2. Calcola il rapporto di accettazione: α = min(1, [π(x')q(x_t|x')] / [π(x_t)q(x'|x_t)])
3. Accetta x_{t+1} = x' con probabilità α, altrimenti x_{t+1} = x_t

La costruzione garantisce che π soddisfi le equazioni di bilancio dettagliato, quindi è la distribuzione stazionaria. L'algoritmo converge a π indipendentemente dalla distribuzione iniziale, sotto condizioni di regolarità.

Gibbs Sampling

Il Gibbs sampling è un caso speciale per distribuzioni multivariate dove si campionano iterativamente le variabili condizionatamente alle altre. Per un vettore x = (x_1, ..., x_d), l'iterazione è:

x_1^{(t+1)} ~ π(x_1 | x_2^{(t)}, ..., x_d^{(t)}) x_2^{(t+1)} ~ π(x_2 | x_1^{(t+1)}, x_3^{(t)}, ..., x_d^{(t)}) ... x_d^{(t+1)} ~ π(x_d | x_1^{(t+1)}, ..., x_{d-1}^{(t+1)})

Se le distribuzioni condizionate sono facilmente campionabili, questo metodo è molto efficiente. È ampiamente usato in modelli grafici probabilistici e nell'inferenza per modelli gerarchici bayesiani.

Hamiltonian Monte Carlo (HMC)

HMC migliora l'efficienza introducendo variabili ausiliarie di momento p e simulando la dinamica Hamiltoniana:

H(x,p) = -log π(x) + (1/2)p^T M^{-1} p

Le equazioni di Hamilton sono: dx/dt = M^{-1}p dp/dt = -∇_x log π(x)

Integrando numericamente queste equazioni (con leapfrog integrator), si propongono stati distanti ma altamente correlati con la distribuzione target, riducendo la correlazione autocorrelazione nella catena. HMC è particolarmente efficace per distribuzioni ad alta dimensionalità ed è implementato in software popolari come Stan e PyMC3.

Applicazioni in Biologia e Genetica

Modelli di Evoluzione del DNA

I processi di Markov sono centrali nella filogenetica per modellare l'evoluzione delle sequenze di DNA lungo alberi evolutivi.

Modello di Jukes-Cantor

Il modello più semplice assume che tutte le sostituzioni nucleotidiche (A, C, G, T) avvengano con lo stesso tasso. Il generatore infinitesimale è:

Q = μ | -3 1 1 1 | | 1 -3 1 1 | | 1 1 -3 1 | | 1 1 1 -3 |

dove μ è il tasso di sostituzione per sito. La probabilità di transizione dopo tempo t è:

$p_{ij}(t) = (1/4) + (3/4)e^{-4\mu t}$ per $i = j$ $p_{ij}(t) = (1/4) - (1/4)e^{-4\mu t}$ per $i \neq j$

Questo modello permette di stimare distanze evolutive tra sequenze e costruire alberi filogenetici.

Modelli più Complessi

Il modello di Kimura (K80) distingue tra transizioni (A↔G, C↔T) e trasversioni (altre sostituzioni). Il modello HKY85 incorpora frequenze nucleotidiche non uniformi. Il modello generale time-reversible (GTR) ha sei parametri di tasso diversi, uno per ogni coppia di nucleotidi.

Modelli di Dinamica di Popolazioni

Processo di Nascita-Morte per Popolazioni

La dimensione di una popolazione può essere modellata come processo di nascita-morte con tassi dipendenti dalla dimensione n:

$\lambda_n = n\lambda$ (tasso di nascita proporzionale) $\mu_n = n\mu + n(n-1)\sigma/2$ (morte + competizione)

dove σ rappresenta la competizione intra-specifica. Per $\lambda > \mu$, la popolazione cresce esponenzialmente in media, ma c'è sempre probabilità positiva di estinzione partendo da popolazioni finite.

Processo di Ramificazione (Branching Process)

Un processo di ramificazione modella popolazioni dove ogni individuo si riproduce indipendentemente. Sia X_n la dimensione della generazione n, e ξ_i il numero di figli dell'individuo i. Allora:

$X_{n+1} = \Sigma_{i=1}^{X_n} \xi_i$

dove $\{\xi_i\}$ sono iid con funzione generatrice $G(s) = E[s^{\xi}]$. La funzione generatrice di X_n è:

$G_n(s) = G_{n-1}(G(s))$

Il comportamento a lungo termine dipende dalla media $m = E[\xi]$:

- Se $m < 1$, la popolazione si estingue quasi certamente
- Se $m = 1$, la popolazione si estingue ancora quasi certamente (tranne caso deterministico)
- Se $m > 1$, c'è probabilità positiva 1-q di sopravvivenza, dove q è la soluzione minima di $q = G(q)$

Quando la popolazione sopravvive con $m > 1$, la dimensione cresce come m^n.

Modelli Epidemiologici SIR

Il modello SIR divide la popolazione in tre compartimenti: Suscettibili (S), Infetti (I), Recuperati (R). In versione stocastica a tempo continuo, le transizioni sono:

$S \to I$ con tasso $\beta SI/N$ (infezione) $I \to R$ con tasso γI (recupero)

dove β è il tasso di contatto efficace, γ il tasso di recupero, e $N = S + I + R$ la popolazione totale. Il numero di riproduzione di base è $R_0 = \beta/\gamma$, rappresentante il numero medio di infezioni secondarie da un individuo infetto in una popolazione completamente suscettibile.

Per $R_0 > 1$, può verificarsi un'epidemia. La dimensione finale dell'epidemia è determinata da equazioni trascendenti derivate dall'analisi del processo. Estensioni includono compartimenti esposti (SEIR), vaccinazione, eterogeneità della popolazione, e network di contatti.

Applicazioni in Fisica e Chimica

Meccanica Statistica e Modello di Ising

Il modello di Ising è un modello di spin su reticolo dove ogni sito i ha spin $\sigma_i \in \{-1, +1\}$. L'Hamiltoniana è:

$$H(\sigma) = -J \sum_{<i,j>} \sigma_i \sigma_j - h \sum_i \sigma_i$$

dove J è l'accoppiamento tra spin vicini, h è il campo magnetico esterno, e la prima somma è su coppie di siti adiacenti.

La dinamica di Glauber definisce una catena di Markov a tempo continuo dove singoli spin flippano con tasso:

$$w(\sigma_i \to -\sigma_i) = \exp(-\beta[H(\sigma^{flip}) - H(\sigma)]/2) / [\exp(-\beta[H(\sigma^{flip}) - H(\sigma)]/2) + \exp(\beta[H(\sigma^{flip}) - H(\sigma)]/2)]$$

dove $\beta = 1/(k_B T)$ è la temperatura inversa. Questa dinamica ha come distribuzione stazionaria la distribuzione di Gibbs:

$$\pi(\sigma) = \exp(-\beta H(\sigma)) / Z(\beta)$$

dove $Z(\beta) = \sum_\sigma \exp(-\beta H(\sigma))$ è la funzione di partizione.

Il modello di Ising esibisce transizioni di fase: per temperatura $T < T_c$ (temperatura critica), si ha magnetizzazione spontanea anche in assenza di campo esterno. La teoria dei processi di Markov fornisce strumenti per studiare l'approssimazione all'equilibrio e le fluttuazioni vicino alla transizione.

Processi di Reazione-Diffusione

I sistemi chimici con molecole di concentrazione bassa richiedono modelli stocastici. Per reazioni come:

$A + B \to C$ con tasso k

il numero di molecole evolve secondo un processo di nascita-morte a tempo continuo. L'algoritmo di Gillespie simula esattamente questa dinamica:

1. Calcola il tasso totale: $a_0 = \Sigma_j a_j$ dove a_j è il tasso della reazione j
2. Tempo fino alla prossima reazione: $\tau \sim \text{Exp}(a_0)$
3. Scegli reazione j con probabilità a_j/a_0
4. Aggiorna lo stato secondo la stechiometria della reazione j

Questo algoritmo Monte Carlo cinetico è esatto per sistemi well-mixed ed è ampiamente usato in biologia dei sistemi e biochimica.

Processi di Lévy e Voli di Lévy in Ecologia

I movimenti animali spesso esibiscono "voli di Lévy": traiettorie con lunghi spostamenti rettilinei alternati a ricerca intensiva locale. Questi sono modellati da processi di Lévy con salti power-law. La distribuzione della lunghezza dei voli è:

$p(l) \sim l^{-\mu}$ per $l \to \infty$

con $1 < \mu \leq 3$. Per $\mu \approx 2$, la strategia è ottimale per ricerca in ambienti con risorse sparse (congettura di Lévy flight foraging).

L'equazione di Fokker-Planck frazionaria descrive la densità di probabilità $p(x,t)$:

$\partial p/\partial t = K_\alpha \partial^\alpha p/\partial|x|^\alpha$

dove $1 < \alpha \leq 2$ è l'indice di Lévy (correlato a μ). Per $\alpha = 2$, si recupera la diffusione normale (browniana).

Applicazioni in Ingegneria e Sistemi

Affidabilità e Manutenzione di Sistemi

I processi di Markov modellano naturalmente sistemi soggetti a guasti e riparazioni. Consideriamo un sistema con n componenti, dove ogni componente può essere funzionante (1) o guasta (0). Lo stato è il vettore $x = (x_1, ..., x_n) \in \{0,1\}^n$.

I tassi di transizione sono:

- Guasto del componente i: λ_i (transizione da stato con $x_i = 1$ a $x_i = 0$)
- Riparazione del componente i: μ_i (transizione da stato con $x_i = 0$ a $x_i = 1$)

La funzione di struttura $\varphi(x) = 1$ se il sistema funziona nello stato x, $\varphi(x) = 0$ altrimenti. L'affidabilità al tempo t è:

$R(t) = \Sigma_{x: \varphi(x)=1} p_x(t)$

dove p_x(t) è la probabilità di essere nello stato x al tempo t.

Per sistemi riparabili, l'availabilità a regime è:

A = lim_{t→∞} Σ_{x: φ(x)=1} p_x(t) = Σ_{x: φ(x)=1} π_x

dove π è la distribuzione stazionaria.

Filtro di Kalman e Stima di Stato

Il filtro di Kalman è un algoritmo ottimale per stimare lo stato di un sistema lineare gaussiano:

x_{t+1} = Ax_t + w_t (equazione di stato) y_t = Cx_t + v_t (equazione di osservazione)

dove w_t ~ N(0, Q) e v_t ~ N(0, R) sono rumori gaussiani indipendenti. Il filtro fornisce ricorsivamente la stima ottimale (nel senso dei minimi quadrati) x̂_t|t e la covarianza dell'errore P_t|t:

Prediction step: x̂_{t+1|t} = Ax̂_{t|t} P_{t+1|t} = AP_{t|t}A^T + Q

Update step: K_{t+1} = P_{t+1|t}C^T(CP_{t+1|t}C^T + R)^{-1} (guadagno di Kalman) x̂_{t+1|t+1} = x̂_{t+1|t} + K_{t+1}(y_{t+1} - Cx̂_{t+1|t}) P_{t+1|t+1} = (I - K_{t+1}C)P_{t+1|t}

Il filtro di Kalman extended (EKF) linearizza sistemi non lineari, mentre il filtro di Kalman unscented (UKF) e il filtro particellare gestiscono non linearità più severe.

Controllo Stocastico Ottimo

Il problema di controllo stocastico considera un sistema dinamico:

dx(t) = f(x(t), u(t), t)dt + g(x(t), t)dW(t)

dove u(t) è il controllo da scegliere per minimizzare il costo:

J = E[∫_0^T L(x(t), u(t), t)dt + Φ(x(T))]

L'equazione di Hamilton-Jacobi-Bellman fornisce condizioni necessarie per l'ottimalità. La funzione valore V(x,t) soddisfa:

-∂V/∂t = min_u [L(x,u,t) + (∂V/∂x)f(x,u,t) + (1/2)Tr(g g^T ∂^2V/∂x^2)]

con condizione terminale V(x,T) = Φ(x). Il controllo ottimale è determinato minimizzando l'espressione tra parentesi quadre.

Per sistemi lineari-quadratici (LQG), il problema ha soluzione esplicita attraverso equazioni di Riccati, separando stima (filtro di Kalman) e controllo (principio di separazione).

Applicazioni in Economia e Scienze Sociali

Modelli di Crescita Economica Stocastica

I modelli di crescita neoclassica incorporano shock stocastici alla produttività. Il modello di base considera un'economia con capitale K(t) che evolve secondo:

$$dK(t) = [sF(K(t)) - \delta K(t)]dt + \sigma K(t)dW(t)$$

dove s è il tasso di risparmio, F è la funzione di produzione, δ è il tasso di deprezzamento, e σW(t) rappresenta shock tecnologici. La distribuzione stazionaria del capitale (se esiste) determina la distribuzione di reddito di lungo periodo.

Modelli di Scelta Discreta Dinamica

Gli agenti economici affrontano sequenze di decisioni discrete (es: lavorare/studiare, sposarsi/rimanere single, comprare/affittare). Rust (1987) modella queste situazioni come MDP dove:

- Stati: caratteristiche dell'agente e ambiente
- Azioni: scelte discrete disponibili
- Transizioni: evoluzione stocastica di caratteristiche
- Payoff: utilità istantanea più valore di continuazione scontato

L'equazione di Bellman è:

$$V(s) = \max_{a \in A} [u(s,a) + \beta \Sigma_{s'} P(s'|s,a)V(s')]$$

dove β è il fattore di sconto temporale. Metodi nested fixed point e MPEC stimano i parametri strutturali da dati osservazionali.

Modelli di Ricerca nel Mercato del Lavoro

Il modello di ricerca di McCall modella la decisione di un disoccupato che riceve offerte salariali w secondo una distribuzione F(w). Ad ogni periodo, decide se accettare l'offerta o rifiutare e continuare a cercare con costo c.

La funzione valore è:

$$V = \max\{w/(1-\beta), -c + \beta \int V \, dF(w)\}$$

Esiste un salario di riserva w* tale che il lavoratore accetta se w ≥ w*. Il salario di riserva soddisfa:

$$w^*/(1-\beta) = -c + \beta \int_{w^*}^{\infty} w/(1-\beta) \, dF(w) + \beta F(w^*) \, w^*/(1-\beta)$$

Questo modello spiega la durata della disoccupazione e l'eterogeneità salariale tra lavoratori simili.

Modelli di Diffusione di Innovazioni

Il processo di Bass modella l'adozione di nuovi prodotti/tecnologie in una popolazione. La probabilità che un individuo adotti al tempo t, dato che non ha ancora adottato, è:

h(t) = p + q N(t)/m

dove p è il coefficiente di innovazione (adozione spontanea), q è il coefficiente di imitazione (adozione per influenza sociale), N(t) è il numero di adottanti al tempo t, e m è il potenziale di mercato.

Il numero di nuovi adottanti segue:

dN(t)/dt = [p + q N(t)/m][m - N(t)]

con soluzione:

N(t) = m [1 - e^(-(p+q)t)] / [1 + (q/p)e^(-(p+q)t)]

Versioni stocastiche incorporano incertezza nei parametri e nell'ambiente competitivo.

Metodi Computazionali Avanzati

Simulazione di Catene di Markov

La simulazione diretta di catene di Markov è fondamentale per analisi numerica quando soluzioni analitiche non sono disponibili.

Metodo della Trasformata Inversa

Per simulare X_{n+1} data X_n = i, generiamo U ~ Uniform(0,1) e settiamo:

X_{n+1} = j se Σ_{k=0}^{j-1} p_{ik} < U ≤ Σ_{k=0}^j p_{ik}

Questo richiede O(|S|) operazioni per transizione. Per catene con molti stati, metodi più efficienti come alias method riducono la complessità a O(1) dopo preprocessing O(|S|).

Metodo di Gillespie per Tempo Continuo

Per processi a tempo continuo con stato i e tassi totali q_i, il tempo fino alla prossima transizione è τ ~ Exp(q_i). La destinazione j è scelta con probabilità q_{ij}/q_i. L'algoritmo è:

1. Genera τ ~ Exp(q_i)
2. Avanza il tempo: t ← t + τ
3. Scegli j con probabilità q_{ij}/q_i
4. Aggiorna stato: i ← j

Questo è esatto per processi di salto puri ed è efficiente quando i tassi non cambiano frequentemente.

Accelerazione e Variance Reduction

Importance Sampling

Per stimare $E_\pi[f]$ quando f ha supporto su regioni di bassa probabilità sotto π, si campiona da una distribuzione alternativa q e si riscala:

$$E_\pi[f] = E_q[f(X)\,\pi(X)/q(X)]$$

La scelta ottimale di q minimizza la varianza dell'estimatore. In MCMC, questo si implementa con sampling preferenziale o con twist della dinamica.

Stratificazione Temporale

Per stimare probabilità di eventi rari in sistemi dinamici (es: rovina, overflow in code, transizioni tra stati metastabili), la stratificazione divide il percorso da stato iniziale a target in livelli intermedi. Si stimano probabilità di transizione tra livelli consecutivi e si moltiplicano.

Se $p = P(A \to B)$, particianando in $A \to C \to B$:

$$p = P(A \to C) \cdot P(C \to B)$$

Con varianza molto ridotta se C è scelto appropriatamente.

Metodi Quasi-Monte Carlo

I metodi quasi-Monte Carlo sostituiscono sequenze aleatorie con sequenze a bassa discrepanza (es: Sobol, Halton), fornendo convergenza $O(\log^d(n)/n)$ invece di $O(1/\sqrt{n})$ per Monte Carlo standard in problemi di integrazione d-dimensionale.

Per MCMC, QMC è più delicato poiché le catene hanno dipendenza temporale. Metodi recenti come Array-RQMC combinano RQMC con catene di Markov, fornendo miglioramenti per problemi specifici.

Teoria Avanzata e Risultati Recenti

Large Deviations e Funzioni di Tasso

La teoria delle grandi deviazioni studia probabilità di eventi rari per processi stocastici. Per una catena di Markov ergodica con distribuzione stazionaria π, la media empirica:

$$L_n = (1/n)\,\Sigma_{k=0}^{n-1}\,\delta_{X_k}$$

(dove δ_x è massa di Dirac in x) converge a π per $n \to \infty$. La teoria delle grandi deviazioni quantifica:

$$P(L_n \approx \mu) \approx e^{(-n\,I(\mu))}$$

dove $I(\mu)$ è la funzione di tasso, caratterizzante quanto è "costoso" osservare distribuzione empirica μ invece di π. Esplicitamente:

$I(\mu) = \sup_f [\Sigma_x \mu(x)f(x) - \log \Sigma_x \pi(x)e^{f(x)}]$

Questa teoria ha applicazioni in fisica statistica (principio di Boltzmann), teoria dell'informazione (codifica), e finanza (probabilità di default).

Convergenza Esponenziale e Gap Spettrale

Per catene di Markov reversibili, la velocità di convergenza alla distribuzione stazionaria è caratterizzata dal gap spettrale:

$\gamma = 1 - \lambda_2$

dove λ_2 è il secondo autovalore più grande (in modulo) della matrice di transizione. Si ha:

$||P^n(i, \cdot) - \pi||_{TV} \leq (1/2)\sqrt{(1/\pi_{min})}\,(1-\gamma)^n$

dove $|| \cdot ||_{TV}$ è la distanza in variazione totale e $\pi_{min} = \min_i \pi_i$. Il tempo di mixing, tempo necessario perché la distanza scenda sotto ε, è:

$t_{mix}(\varepsilon) \approx (1/\gamma) \log(1/(\varepsilon\pi_{min}))$

Per catene su grafi, γ è legato alla conducibilità (Cheeger inequality), fornendo metodi geometrici per stimare convergenza.

Catene di Markov su Gruppi e Rappresentazioni

Catene di Markov su gruppi finiti (es: permutazioni, matrici) hanno struttura aggiuntiva sfruttabile. Il cut-off phenomenon afferma che per molte catene naturali su gruppi, la convergenza avviene bruscamente: la distanza dalla stazionaria rimane vicina a 1 per $t < t_{mix}$ e cade rapidamente a 0 per $t > t_{mix}$.

La teoria delle rappresentazioni fornisce strumenti per analizzare queste catene. La decomposizione spettrale usa rappresentazioni irriducibili del gruppo per diagonalizzare la matrice di transizione.

Per il riffle shuffle (mescolamento standard di carte), servono $(3/2)\log_2(52) \approx 8.5$ shuffle per mescolare un mazzo da 52 carte. Dopo 7 shuffle, le carte sono ancora significativamente ordinate; dopo 9, sono ben mescalate.

Processi di Markov Interagenti

I sistemi di particelle interagenti consistono di molti processi di Markov accoppiati. Il modello del votante, il processo di contatto, e il modello di esclusione asimmetrico sono esempi classici.

Nel processo di esclusione semplice simmetrico (SSEP), particelle occupano siti di un reticolo con al più una particella per sito. Coppie di particelle su siti adiacenti scambiano posizioni con tasso 1. Questo modello è esattamente risolubile e mostra comportamento idrodinamico: al limite di scala diffusiva, la densità locale soddisfa l'equazione del calore.

Il teorema di dualità collega sistemi con molte particelle a sistemi con poche particelle, facilitando l'analisi. Per SSEP, il processo duale è ancora SSEP con ruolo di particelle e vuoti scambiato.

Sistemi di Salto di Piecewise Deterministici (PDMP)

I PDMP combinano evoluzione deterministica con salti aleatori. Tra salti, il sistema evolve secondo un'ODE:

dx/dt = f(x)

I salti avvengono con tassi dipendenti dallo stato λ(x), e la destinazione dopo un salto da x è determinata da una distribuzione Q(x, dy).

I PDMP includono come casi speciali processi di salto puri (quando f = 0) e processi di diffusione (limite di salti frequenti piccoli). Sono usati per modellare sistemi ibridi, reti neurali spiking, e sistemi biologici con eventi discreti.

Il generatore esteso opera su funzioni dello stato e del tempo dall'ultimo salto:

$Lφ(x,τ) = f(x)\cdot\nabla φ + \partial φ/\partial τ + λ(x)[\int φ(y,0)Q(x,dy) - φ(x,τ)]$

Frontiere di Ricerca Contemporanea

Machine Learning e Processi di Markov

Generative Adversarial Networks (GANs) e Dinamica di Nash

Il training di GANs può essere visto come un gioco tra generatore e discriminatore che cercano un equilibrio di Nash. La dinamica è:

θ_G^{(t+1)} = θ_G^{(t)} + α ∇*{θ_G}* *E*{z}[log(1 - D(G(z)))] θ_D^{(t+1)} = θ_D^{(t)} + α ∇_{θ_D} E_x[log D(x)] + E_z[log(1 - D(G(z)))]

Questa dinamica può essere analizzata come catena di Markov in spazio dei parametri, anche se non ergodica in generale. La convergenza a equilibri e la presenza di mode collapse sono attive aree di ricerca.

Diffusion Models

I modelli di diffusione sono recentemente emersi come metodo potente per generazione di immagini. Consistono di:

1. Forward process: q(x_t|x_0) che progressivamente aggiunge rumore gaussiano fino a x_T ~ N(0,I)
2. Reverse process: p_θ(x_{t-1}|x_t) che impara a "denoising" per generare da rumore

Il forward process è tipicamente:

$x_t = \sqrt{(\alpha_t)}\, x_0 + \sqrt{(1-\alpha_t)}\, \varepsilon,\ \varepsilon \sim N(0,I)$

con α_t decrescente. Il reverse process impara a predire il rumore ε per invertire il processo. Questo è essenzialmente un processo di diffusione a tempo discreto, e la teoria dei processi di Markov fornisce fondamenti per analisi di convergenza e proprietà di generazione.

Quantum Computing e Catene Quantistiche di Markov

Le catene quantistiche di Markov generalizzano catene classiche a sistemi quantistici. Lo stato è una matrice densità ρ e l'evoluzione è data da mappe completamente positive e trace-preserving (CPTP):

$\rho_{t+1} = \Phi(\rho_t)$

dove Φ ha rappresentazione di Kraus: $\Phi(\rho) = \Sigma_k K_k \rho K_k^\dagger$ con $\Sigma_k K_k^\dagger K_k = I$.

L'analogo quantistico della distribuzione stazionaria è uno stato stazionario ρ^* tale che $\Phi(\rho^*) = \rho^*$. La convergenza e l'ergodicità quantistica sono definite attraverso la distanza trace: $||\rho - \sigma||_1 = Tr\sqrt{((\rho-\sigma)^\dagger(\rho-\sigma))}$.

Algoritmi quantistici come quantum walks sono analoghi quantistici di random walks e mostrano speed-up per problemi come ricerca su grafi. Il quantum PageRank generalizza PageRank classico a network quantistici.

Mean Field Games e Sistemi con Infiniti Agenti

I mean field games studiano equilibri di Nash per popolazioni con infiniti agenti identici che interagiscono attraverso la distribuzione aggregata. Ogni agente risolve un problema di controllo ottimale:

$dX_t = u_t\, dt + \sigma\, dW_t \quad \min_u E[\int_0^T L(X_t, u_t, \mu_t)\, dt + \Phi(X_T, \mu_T)]$

dove μ_t è la distribuzione degli stati di tutti gli agenti al tempo t. In equilibrio, μ_t evolve secondo l'equazione di Fokker-Planck con drift dato dal controllo ottimale.

Il sistema accoppiato di equazioni (Hamilton-Jacobi-Bellman per agenti + Fokker-Planck per distribuzione) determina l'equilibrio. Questi modelli sono applicati in economia (competizione tra firme), finanza (trade ottimale con impatto di mercato), e ingegneria (controllo di sciami).

Reinforcement Learning Profondo e Approssimazioni

Il successo recente di RL (AlphaGo, GPT, robotica) deriva dall'uso di approssimazioni neurali per funzioni valore. Policy gradient methods ottimizzano direttamente:

$J(\theta) = E_{\tau\sim\pi_\theta}[\Sigma_t r_t]$

con gradiente:

$\nabla\theta\, J(\theta) = E\{\tau\sim\pi_\theta\}[\Sigma_t\, \nabla\theta \log \pi_\theta(a_t|s_t)\, \Sigma\{t'\geq t\}\, r_{t'}]$

Actor-Critic combina questo con approssimazione della funzione valore. Proximal Policy Optimization (PPO) stabilizza il training con un constraint:

max_θ E[min(r_t(θ)A_t, clip(r_t(θ), 1-ε, 1+ε)A_t)]

dove r_t(θ) = π_θ(a|s)/π_{θ_old}(a|s) e A_t è l'advantage. La teoria della convergenza per questi metodi su spazi di stati/azioni enormi è ancora in sviluppo.

Processi di Markov e Causalità

L'inferenza causale studia relazioni causa-effetto da dati osservazionali o sperimentali. I modelli causali strutturali rappresentano meccanismi causali come grafi diretti aciclici (DAGs) con equazioni strutturali.

I processi di Markov temporali forniscono framework naturale per causalità dinamica. Granger causality definisce X causa Y se passati valori di X migliorano la predizione di Y oltre a passati valori di Y stesso:

Y non Granger-causa X se P(X_t | X_{t-1}, ..., Y_{t-1}, ...) = P(X_t | X_{t-1}, ...)

Per processi di Markov, questo si semplifica usando la proprietà di Markov. Transfer entropy quantifica il trasferimento di informazione:

T_{Y→X} = Σ p(x_t, x_{t-1}, y_{t-1}) log [p(x_t|x_{t-1},y_{t-1})/p(x_t|x_{t-1})]

Questi concetti sono applicati in neuroscienze (connettività cerebrale), economia (lead-lag relationships), e climate science.

Conclusioni e Prospettive Future

I processi di Markov hanno rivoluzionato la modellistica matematica di sistemi stocastici dinamici attraverso la loro eleganza teorica e vastissima applicabilità. La proprietà fondamentale di assenza di memoria, pur essendo una semplificazione della realtà, cattura l'essenza di innumerevoli fenomeni naturali e artificiali.

Gli sviluppi recenti mostrano direzioni promettenti:

Scalabilità Computazionale: L'integrazione con machine learning e computing ad alte prestazioni permette di affrontare problemi con spazi di stati enormi precedentemente intrattabili. Metodi come deep reinforcement learning e variational inference scalano a milioni di stati.

Sistemi Multi-Scala: La comprensione di come comportamenti microscopici (molecolari, individuali) emergono in fenomeni macroscopici (termodinamici, sociali) attraverso limiti idrodinamici e mean field è area attiva di ricerca.

Non-Markovianità: Mentre la proprietà di Markov è potente, molti sistemi reali hanno memoria. Teorie di processi semi-Markoviani, processi frazionari, e memoria kernel estendono il framework classico.

Quantum e Stocastico: L'intersezione di meccanica quantistica e processi stocastici apre nuove frontiere, da quantum walks a quantum machine learning, con potenziali applicazioni in crittografia, simulazione, e ottimizzazione.

Etica e Società: L'uso crescente di MDP e RL in decisioni che impattano vite umane (medicina, giustizia, finanza) richiede sviluppo di metodi che incorporano fairness, interpretabilità, e robustezza a distributional shift.

La teoria dei processi di Markov continua ad evolversi, guidata sia da sfide teoriche profonde sia da applicazioni pratiche urgenti. La sua unificazione di idee da probabilità, analisi, algebra, e fisica garantisce che rimarrà area centrale della matematica applicata per decenni a venire.

Processi di Markov a Salti con Misure di Lévy

Fondamenti dei Processi di Lévy

Un processo di Lévy {X(t), t ≥ 0} è un processo stocastico a tempo continuo con le seguenti proprietà:

1. X(0) = 0 quasi certamente
2. **Incrementi indipendenti**: X(t) - X(s) è indipendente da {X(u), u ≤ s} per ogni t > s
3. **Incrementi stazionari**: La distribuzione di X(t+h) - X(t) dipende solo da h, non da t
4. **Continuità stocastica**: lim_{h→0} P(|X(t+h) - X(t)| > ε) = 0 per ogni ε > 0

I processi di Lévy generalizzano sia il moto browniano (traiettorie continue) sia i processi di Poisson composti (salti puri), permettendo di modellare sia componenti diffusive sia discontinuità.

Decomposizione di Lévy-Khintchine

La caratteristica fondamentale dei processi di Lévy è che la loro funzione caratteristica ha forma esponenziale. Per ogni processo di Lévy X(t), esiste una tripla (a, σ², ν) chiamata tripla di Lévy tale che:

E[e^{iθX(t)}] = exp(tΨ(θ))

dove Ψ(θ) è l'esponente di Lévy:

Ψ(θ) = iaθ - (σ²θ²)/2 + ∫*{R{0}} [e^{iθx} - 1 - iθx·1*{|x|<1}] ν(dx)

I tre parametri hanno interpretazioni specifiche:

- **a ∈ R**: drift deterministico
- **σ² ≥ 0**: intensità della componente browniana (diffusione gaussiana)
- **ν**: misura di Lévy (o misura di salto), una misura σ-finita su R{0} che soddisfa:

$\int_{R\{0\}} \min(1, x^2)\, \nu(dx) < \infty$

La misura di Lévy ν caratterizza completamente la struttura dei salti del processo. L'integrale di ν su un insieme A rappresenta il tasso atteso di salti con dimensione in A per unità di tempo.

Rappresentazione di Lévy-Itô

Ogni processo di Lévy può essere decomposto come somma di tre componenti indipendenti:

$X(t) = at + \sigma W(t) + X^{small}(t) + X^{large}(t)$

dove:

1. **at**: componente di drift lineare
2. **$\sigma W(t)$**: moto browniano standard con volatilità σ
3. **$X^{small}(t) = \int_0^t \int_{|x|<1} x[N(ds,dx) - ds\nu(dx)]$**: somma compensata di salti piccoli ($|x| < 1$)
4. **$X^{large}(t) = \int_0^t \int_{|x|\geq 1} x\, N(ds,dx)$**: somma di salti grandi ($|x| \geq 1$)

dove $N(ds,dx)$ è una misura aleatoria di Poisson su $[0,\infty) \times (R\{0\})$ con misura di intensità $ds \cdot \nu(dx)$.

Questa decomposizione mostra che ogni processo di Lévy consiste di:

- Movimento deterministico (drift)
- Diffusione browniana
- Accumulo di infiniti salti piccoli
- Salti grandi discreti

Esempi Fondamentali di Processi di Lévy

Processo di Poisson Composto

Il più semplice processo di Lévy con salti ha la forma:

$X(t) = \Sigma_{i=1}^{N(t)} Y_i$

dove $N(t)$ è un processo di Poisson con intensità λ e $\{Y_i\}$ sono variabili aleatorie iid con distribuzione F, indipendenti da $N(t)$. La misura di Lévy è:

$\nu(dx) = \lambda F(dx)$

e l'esponente di Lévy è:

$\Psi(\theta) = \lambda \int [e^{i\theta x} - 1] F(dx)$

Processo Gamma

Il processo gamma Gamma(t; a, b) ha incrementi che seguono distribuzioni gamma. È un processo crescente (no salti negativi) con misura di Lévy:

v(dx) = (a/x)e^{-bx} dx per x > 0

e parametri della tripla di Lévy: drift a = ∫_0^1 x v(dx), σ² = 0. È usato per modellare tempi di attesa, intensità di sinistri in assicurazioni, e volatilità stocastica.

Processo di Variance Gamma (VG)

Il processo VG è la differenza di due processi gamma indipendenti:

X(t) = Gamma^+(t; a_+, b_+) - Gamma^-(t; a_-, b_-)

Ha misura di Lévy:

v(dx) = (C/|x|)e^{-M|x|+θx} dx

dove C, M, θ sono parametri. È usato in finanza per modellare rendimenti di asset con code grasse e asimmetria.

Processo Normal Inverse Gaussian (NIG)

Il processo NIG ha misura di Lévy:

v(dx) = (αδ/π|x|) K_1(α|x|)e^{βx} dx

dove K_1 è la funzione di Bessel modificata di secondo tipo, e α, β, δ sono parametri. Combina code grasse con decadimento semi-esponenziale ed è molto usato in finanza per la sua flessibilità.

Processo α-Stabile

I processi α-stabili generalizzano il moto browniano permettendo code più pesanti. Per 0 < α < 2, hanno misura di Lévy:

v(dx) = [C_+ x^{-(1+α)} 1_{x>0} + C_- |x|^{-(1+α)} 1_{x<0}] dx

Per α < 2, la varianza è infinita, catturando eventi estremi. Il caso α = 2 corrisponde al moto browniano, α = 1 alla distribuzione di Cauchy.

Generatore Infinitesimale per Processi di Lévy

Per un processo di Lévy X(t), il generatore infinitesimale agisce su funzioni sufficientemente regolari f secondo:

Lf(x) = af'(x) + (σ²/2)f''(x) + ∫*{R{0}} [f(x+y) - f(x) - yf'(x)1*{|y|<1}] v(dy)

Questo operatore integro-differenziale generalizza il generatore dei processi di diffusione includendo il termine integrale che cattura l'effetto dei salti. Per funzioni $f \in C^2_b$ (limitate con derivate seconde limitate), vale:

$d/dt\ E[f(X(t))] = E[Lf(X(t))]$

Equazioni Integro-Differenziali

Per processi di Lévy, le equazioni di Kolmogorov diventano equazioni integro-differenziali (PIDE - Partial Integro-Differential Equations).

L'equazione di Kolmogorov backward per $u(x,t) = E[g(X(T)) \mid X(t) = x]$ è:

$\partial u/\partial t + Lu = 0$

con condizione terminale $u(x,T) = g(x)$. Esplicitamente:

$\partial u/\partial t + a\partial u/\partial x + (\sigma^2/2)\partial^2 u/\partial x^2 + \int_{R\{0\}} [u(x+y,t) - u(x,t) - y(\partial u/\partial x)1_{\{|y|<1\}}]\, v(dy) = 0$

L'equazione di Kolmogorov forward (Fokker-Planck) per la densità $p(x,t)$ è:

$\partial p/\partial t = -\partial(ap)/\partial x + (\sigma^2/2)\partial^2 p/\partial x^2 + \int_{R\{0\}} [p(x-y,t)v(dy) - p(x,t)v(R\{0\})]$

Queste PIDE sono più complesse delle PDE per diffusioni pure e richiedono metodi numerici specializzati per la soluzione.

Subordinazione

Un concetto potente nella teoria dei processi di Lévy è la subordinazione: costruire un nuovo processo di Lévy cambiando la scala temporale con un subordinatore indipendente.

Un **subordinatore** $T(t)$ è un processo di Lévy crescente ($T(t) \geq T(s)$ per $t \geq s$). Se $X(t)$ è un processo di Lévy e $T(t)$ è un subordinatore indipendente, allora:

$Y(t) = X(T(t))$

è ancora un processo di Lévy (il processo subordinato).

Esempi:

1. **Tempo di Variance Gamma**: Se $X(t)$ è moto browniano con drift e $T(t)$ è processo gamma, $Y(t) = X(T(t))$ è il processo variance gamma

2. **Normal Inverse Gaussian**: Se $X(t)$ è moto browniano con drift e $T(t)$ è processo inverse Gaussian, $Y(t)$ è processo NIG

La subordinazione permette di costruire processi con proprietà desiderate combinando processi più semplici.

Applicazioni in Finanza

I processi di Lévy sono cruciali nella finanza moderna per superare le limitazioni del modello di Black-Scholes.

Modelli di Prezzi con Salti

Il prezzo di un asset può essere modellato come:

$S(t) = S(0) \exp(X(t))$

dove $X(t)$ è un processo di Lévy. Sotto la misura risk-neutral Q, l'esponente di Lévy deve soddisfare la condizione di no-arbitraggio:

$E^Q[S(t)/S(0)] = e^{rt}$

dove r è il tasso privo di rischio, implicando:

$\Psi(-i) = r$

Pricing di Opzioni con Trasformate di Fourier

Per opzioni europee su sottostanti modellati con processi di Lévy, i prezzi possono essere calcolati efficientemente usando trasformate di Fourier. Il prezzo di una call è:

$C(K,T) = e^{-rT} \int_K^\infty (S - K)p(S,T)dS$

dove $p(S,T)$ è la densità risk-neutral. Usando trasformate di Fourier:

$C(K,T) = e^{-rT}/\pi \int_0^\infty Re[e^{-iuK}\varphi(u-i)/(-iu(1+iu))]du$

dove $\varphi(u) = E[e^{iuX(T)}]$ è la funzione caratteristica. Questo metodo (FFT pricing) è molto più efficiente del Monte Carlo per processi di Lévy.

Calibrazione ai Dati di Mercato

I modelli di Lévy possono essere calibrati a volatility smile osservate nei mercati. Per esempio, il modello VG ha parametri (σ, ν, θ) che controllano:

- σ: volatilità browniana
- ν: grado di attività dei salti (kurtosis)
- θ: asimmetria (skewness)

Questi possono essere stimati minimizzando l'errore tra prezzi di opzioni teorici e di mercato.

Problemi di Controllo Ottimo con Processi di Lévy

I problemi di controllo stocastico con processi di Lévy generalizzano quelli con diffusioni. Il problema è:

max_u E[∫_0^T L(X(t), u(t))dt + Φ(X(T))]

soggetto a:

dX(t) = b(X(t), u(t))dt + σ(X(t), u(t))dW(t) + ∫_{R{0}} c(X(t-), u(t), z)[N(dt,dz) - v(dz)dt]

dove N è misura aleatoria di Poisson. L'equazione di Hamilton-Jacobi-Bellman diventa una PIDE:

-∂V/∂t = max_u {L(x,u) + b(x,u)V_x + (σ²(x,u)/2)V_{xx} + ∫[V(x+c(x,u,z)) - V(x) - c(x,u,z)V_x 1_{|z|<1}]v(dz)}

con condizione terminale V(x,T) = Φ(x). La soluzione richiede tecniche numeriche avanzate come finite differences, finite elements, o metodi di viscosità.

Teoria Ergodica e Teoremi Limite

Ergodicità e Mixing

Un processo di Markov si dice ergodico se ha una unica distribuzione stazionaria π e se, partendo da qualsiasi distribuzione iniziale, la distribuzione al tempo t converge a π per t → ∞. La proprietà di mixing è più forte e richiede che la correlazione tra eventi separati temporalmente decada a zero:

lim_{t→∞} P(X(t) ∈ A, X(0) ∈ B) = π(A)P(X(0) ∈ B)

per insiemi misurabili A e B.

Per catene di Markov irriducibili e aperiodiche su spazi finiti, l'ergodicità è automatica. Per spazi infiniti, sono necessarie condizioni aggiuntive come la ricorrenza positiva. Il tasso di convergenza alla distribuzione stazionaria è caratterizzato dal gap spettrale, ovvero la differenza tra 1 (il primo autovalore) e il modulo del secondo autovalore della matrice di transizione.

Teoremi Limite Centrali per Processi di Markov

Il teorema del limite centrale classico afferma che la somma normalizzata di variabili aleatorie indipendenti converge in distribuzione a una gaussiana. Per processi di Markov, esistono versioni analoghe sotto appropriate condizioni di dipendenza.

Per una catena di Markov ergodica con distribuzione stazionaria π, consideriamo una funzione f con valore atteso stazionario μ = Σ_i π_i f(i). Sotto condizioni di regolarità, vale:

(1/√n) Σ_{k=0}^{n-1} [f(X_k) - μ] →^d N(0, σ²)

dove la varianza asintotica σ² dipende non solo dalla varianza di f sotto π, ma anche dalle correlazioni temporali nel processo:

σ² = Var_π(f) + 2Σ_{k=1}^∞ Cov_π(f(X_0), f(X_k))

dove $\mathrm{Var}_\pi(f)$ è la varianza di f rispetto alla distribuzione stazionaria π, e $\mathrm{Cov}_\pi(f(X_0), f(X_k))$ sono le autocovarianze temporali.

Questa formula mostra che la varianza asintotica include sia la varianza intrinseca della funzione f sia l'effetto cumulativo delle correlazioni temporali nel processo. Se le autocovarianze sono positive (correlazione positiva nel tempo), σ^2 sarà maggiore di $\mathrm{Var}_\pi(f)$; se sono negative, sarà minore.

Per processi con mixing rapido, le autocovarianze decadono velocemente e la somma converge. Per catene reversibili, esistono espressioni alternative che facilitano il calcolo di σ^2:

$$\sigma^2 = \mathrm{Var}_\pi(f) + 2\sum_{k=1}^{\infty} \langle f - \mu, P^k(f - \mu) \rangle_\pi$$

dove $\langle \cdot, \cdot \rangle_\pi$ denota il prodotto scalare nello spazio $L^2(\pi)$. Il gap spettrale $\gamma = 1 - |\lambda_2|$ controlla il decadimento delle autocovarianze e quindi la velocità di mixing del processo.

Condizioni di Applicabilità

Il teorema del limite centrale per catene di Markov richiede condizioni più stringenti rispetto al caso di variabili indipendenti:

1. **Ergodicità**: La catena deve essere ergodica (irriducibile, aperiodica, ricorrente positiva)
2. **Momento secondo finito**: $E_\pi[f^2] < \infty$
3. **Convergenza della serie**: $\sum_{k=1}^{\infty} |\mathrm{Cov}_\pi(f(X_0), f(X_k))| < \infty$

Quest'ultima condizione è automaticamente soddisfatta per catene con mixing esponenziale, dove le autocovarianze decadono come $O(e^{-\gamma k})$ per qualche $\gamma > 0$.

Applicazioni Pratiche

Il teorema del limite centrale per processi di Markov ha importanti applicazioni pratiche:

Intervalli di confidenza per MCMC: Quando si usa Monte Carlo via catene di Markov per stimare $E_\pi[f]$, la stima $(1/n)\sum_{k=1}^{n} f(X_k)$ è approssimativamente normale con media μ e varianza σ^2/n, permettendo la costruzione di intervalli di confidenza.

Analisi di prestazioni: In teoria delle code e sistemi dinamici, il teorema giustifica l'uso di approssimazioni normali per medie temporali di quantità di interesse.

Test statistici: Permette di costruire test di ipotesi per proprietà di processi stocastici basati su osservazioni temporali.

La stima della varianza asintotica σ^2 da dati è un problema non triviale, richiedendo metodi come batch means, overlapping batch means, o spectral variance estimation.

Convergenza Debole e Teoremi di Invarianza

Introduzione ai Teoremi di Invarianza

I teoremi di invarianza (o teoremi limite funzionali) estendono i teoremi limite classici (legge dei grandi numeri, teorema del limite centrale) allo spazio delle funzioni. Invece di considerare la convergenza di variabili aleatorie scalari, studiano la convergenza di interi processi stocastici visti come traiettorie in spazi funzionali.

Spazi Metrici per Processi Stocastici

Spazio di Skorohod D[0,1]

Lo spazio di Skorohod D[0,1] è lo spazio delle funzioni càdlàg (continue a destra con limiti a sinistra) su [0,1]. È lo spazio naturale per studiare processi con salti.

Una funzione $f \in D[0,1]$ soddisfa:

- $f(t+) = f(t)$ per ogni $t \in [0,1)$ (continuità a destra)
- $f(t-) = \lim_{s\uparrow t} f(s)$ exists per ogni $t \in (0,1]$ (esistenza limite sinistro)

La **topologia di Skorohod** è generata dalla metrica:

$$d(f,g) = \inf_{\lambda\in\Lambda} \{||\lambda - id||_\infty \vee ||f - g\circ\lambda||_\infty\}$$

dove Λ è l'insieme delle funzioni $\lambda: [0,1] \to [0,1]$ strettamente crescenti con $\lambda(0) = 0$, $\lambda(1) = 1$.

Questa metrica permette di confrontare funzioni modulo "piccole deformazioni temporali", essenziale per processi con salti.

Spazio di Wiener C[0,1]

Lo spazio C[0,1] delle funzioni continue è un sottospazio chiuso di D[0,1]. La topologia indotta è quella della convergenza uniforme:

$$d(f,g) = \sup_{t\in[0,1]} |f(t) - g(t)|$$

Questo è lo spazio naturale per processi con traiettorie continue come il moto browniano.

Convergenza Debole in Spazi Funzionali

Definizione di Convergenza Debole

Una sequenza di misure di probabilità $\{P_n\}$ su $(D[0,1], B(D[0,1]))$ converge debolmente a P (scritto $P_n \Rightarrow P$) se:

$$\int f\, dP_n \to \int f\, dP$$

per ogni funzione continua limitata $f: D[0,1] \to R$.

Equivalentemente, per processi stocastici $\{X_n\}$ e X:

$X_n \Rightarrow X$ (in distribuzione) se $L(X_n) \Rightarrow L(X)$

dove $L(\cdot)$ denota la legge (distribuzione).

Criteri di Tightness

Il **teorema di Prokhorov** afferma che convergenza debole equivale a:

1. Tightness: Per ogni $\varepsilon > 0$ esiste compatto K_ε tale che $P_n(K_\varepsilon) > 1 - \varepsilon$ per ogni n
2. Convergenza delle distribuzioni finito-dimensionali

Per processi su $D[0,1]$, il **criterio di Aldous** fornisce condizioni verificabili di tightness basate su moduli di continuità.

Teorema (Aldous): $\{X_n\}$ è tight in $D[0,1]$ se:

1. Per ogni $\eta > 0$, $\sup_n P(|X_n(0)| > M) \to 0$ quando $M \to \infty$

2. Per ogni $\varepsilon, \eta > 0$, esiste $\delta > 0$ tale che:

$$\lim \sup_n \sup_{\tau,\theta\leq\delta} P(|X_n(\tau+\theta) - X_n(\tau)| > \varepsilon) < \eta$$

dove il sup è su tempi di arresto $\tau \leq 1-\delta$.

Teorema di Donsker (Principio di Invarianza)

Formulazione Classica

Il teorema di Donsker è il teorema del limite centrale funzionale più fondamentale.

Teorema (Donsker, 1951): Siano $\{\xi_i\}_{i\geq1}$ variabili aleatorie iid con $E[\xi_i] = 0$ e $Var[\xi_i] = \sigma^2 < \infty$. Definiamo:

$S_n(t) = (1/\sqrt{n}) \Sigma_{i=1}^{\lfloor nt \rfloor} \xi_i$ per $t \in [0,1]$

esteso linearmente tra punti. Allora:

$S_n \Rightarrow \sigma W$ in $D[0,1]$

dove W è moto browniano standard.

Interpretazione: Somme parziali normalizzate di variabili iid convergono a moto browniano come processi stocastici, non solo nei valori finito-dimensionali.

Estensione a Processi Interpolati

Una versione più raffinata usa interpolazione lineare continua:

$\tilde{S}_n(t) = S_n(\lfloor nt \rfloor) + (nt - \lfloor nt \rfloor)(\xi_{\lfloor nt \rfloor+1})$

Allora $\tilde{S}_n \Rightarrow \sigma W$ in $C[0,1]$, convergenza nello spazio delle funzioni continue.

Principi di Invarianza per Catene di Markov

Teorema di Donsker per Catene Ergodiche

Teorema: Sia $\{X_n\}$ catena di Markov ergodica con distribuzione stazionaria π e $f: S \to \mathbb{R}$ con $E_\pi[f] = 0$, $E_\pi[f^2] < \infty$. Definiamo:

$S_n(t) = (1/\sqrt{n}) \sum_{k=0}^{\lfloor nt \rfloor} f(X_k)$

Se la varianza asintotica $\sigma^2 = \mathrm{Var}_\pi(f) + 2\sum_{k=1}^{\infty} \mathrm{Cov}_\pi(f(X_0), f(X_k))$ è finita e positiva, allora:

$S_n \Rightarrow \sigma W$ in $D[0,1]$

Condizioni sufficienti:

- Mixing esponenziale: $|\mathrm{Cov}_\pi(f(X_0), f(X_k))| \leq C e^{-\gamma k}$
- Reversibilità con gap spettrale $\gamma > 0$

Applicazioni del Principio di Invarianza

Intervalli di confidenza funzionali: Data una traiettoria lunga $\{X_0, ..., X_N\}$ di una catena ergodica, possiamo costruire bande di confidenza per la media cumulativa:

$(1/n) \sum_{k=0}^{n} f(X_k) \approx \mu + (\sigma/\sqrt{n}) W(n/N)$

dove W è browniano standard. Questo permette visualizzazione grafica dell'incertezza nella stima.

Test di stazionarietà: Il principio di invarianza giustifica test basati su statistiche funzionali come:

$T_n = \sup_{t \in [0,1]} |S_n(t)|$

che sotto H_0 converge in distribuzione a $\sup_{t \in [0,1]} |\sigma W(t)|$, la cui distribuzione è nota (distribuzione di Kolmogorov-Smirnov).

Convergenza a Processi di Diffusione

Scaling Limits di Catene di Markov

Catene di Markov con transizioni piccole e frequenti convergono, sotto scaling appropriato, a processi di diffusione.

Setup: Consideriamo catena $\{X_n^\varepsilon\}$ con transizioni:

$$X_{n+1}^\varepsilon = X_n^\varepsilon + \varepsilon b(X_n^\varepsilon) + \sqrt{\varepsilon}\,\sigma(X_n^\varepsilon)\xi_{n+1} + o(\sqrt{\varepsilon})$$

dove $\{\xi_n\}$ sono iid con media 0, varianza 1.

Definiamo processo continuo $X^\varepsilon(t) = X_{\lfloor t/\varepsilon \rfloor}^\varepsilon$.

Teorema: Sotto condizioni di regolarità su b e σ:

$$X^\varepsilon \Rightarrow X \text{ in } D[0,\infty)$$

dove X soddisfa l'equazione differenziale stocastica:

$$dX(t) = b(X(t))dt + \sigma(X(t))dW(t)$$

Metodo dei Generatori

La convergenza può essere dimostrata mostrando che il generatore L^ε della catena rescaled converge al generatore L della diffusione:

$$L^\varepsilon f(x) = E[f(X_1^\varepsilon) - f(X_0^\varepsilon) \mid X_0^\varepsilon = x]/\varepsilon \to Lf(x) = b(x)f'(x) + (\sigma^2(x)/2)f''(x)$$

per funzioni test $f \in C_c^\infty$.

Esempio: Random walk semplice simmetrico rescaled:

$X_n^\varepsilon = \varepsilon S_n$ dove $S_n = \Sigma_{i=1}^n \xi_i$, $\xi_i \in \{-1,+1\}$

$$X^\varepsilon(t) = \varepsilon S_{\lfloor t/\varepsilon^2 \rfloor}$$

converge a moto browniano: $X^\varepsilon \Rightarrow W$ quando $\varepsilon \to 0$.

Teoremi di Invarianza per Processi di Punto

Convergenza di Processi di Conteggio

Teorema: Sia $\{N_n(t), t \geq 0\}$ sequenza di processi di conteggio con intensità $\lambda_n(t)$. Se:

$\lambda_n(t) \to \lambda(t)$ uniformemente su compatti

allora sotto condizioni di regolarità:

$N_n \Rightarrow N$

dove N è processo di Poisson non-omogeneo con intensità $\lambda(t)$.

Processi di Rinnovo Generalizzati

Per processi di rinnovo con tempi tra arrivi {T_i^n}:

Teorema: Se T_i^n ⇒ T in distribuzione e E[T_i^n] → E[T], allora il processo di rinnovo centrato:

R_n(t) = √n [N_n(t)/n - t/E[T]]

converge a processo gaussiano con funzione di covarianza specifica dipendente dalla distribuzione limite di T.

Applicazioni alle Catene di Markov Monte Carlo

Convergenza di MCMC a Stazionarietà

Consideriamo catena ergodica che converge a π. Il processo:

Z_n(t) = √n [π_n(⌊nt⌋) - π]

dove π_n(k) è distribuzione empirica fino al tempo k, converge a processo gaussiano:

Z_n ⇒ Z in D[0,1]

dove Z è diffusione con drift e varianza determinati dal gap spettrale e autofunzioni della catena.

Diagnostica di Convergenza

Il principio di invarianza giustifica metodi grafici per diagnosticare convergenza:

Trace plots: Visualizzare S_n(t) e verificare comportamento browniano **Effective sample size**: Stimare σ² dalla serie temporale e calcolare n_eff = n σ_0²/σ² **Multiple chains**: Confrontare traiettorie da starts diversi, devono convergere allo stesso processo limite

Convergenza di Processi Empirici

Teorema di Glivenko-Cantelli Funzionale

Per dati iid {X_1, ..., X_n} con CDF F, la funzione di distribuzione empirica:

F_n(x) = (1/n) Σ_{i=1}^n 1_{X_i ≤ x}

converge uniformemente: sup_x |F_n(x) - F(x)| → 0 quasi certamente.

Processo Empirico

Il processo empirico rescaled:

$G_n(x) = \sqrt{n}\,[F_n(x) - F(x)]$

converge in distribuzione a ponte browniano:

$G_n \Rightarrow B^0$ in $D(-\infty,\infty)$

dove B^0 è processo gaussiano con $E[B^0(x)] = 0$ e $\mathrm{Cov}(B^0(x), B^0(y)) = F(\min(x,y)) - F(x)F(y)$.

Per catene di Markov, versioni analoghe richiedono mixing conditions.

Metodi di Dimostrazione

Metodo delle Martingale

Molti teoremi di invarianza si dimostrano costruendo approssimazioni martingala. Se:

$M_n(t) = S_n(t) - \int_0^t b_n(s)ds$

è martingala (o quasi-martingala) con variazione quadratica prevedibile:

$\langle M_n \rangle(t) \to \sigma^2 t$

allora $M_n \Rightarrow \sigma W$.

Metodo dei Generatori Estesi

Per processi di Markov, si mostra che per ogni $f \in C_0^\infty$:

$E[f(X_n(t))] - f(X_n(0)) - \int_0^t L_n f(X_n(s))ds \to 0$

e $L_n \to L$ dove L è generatore del processo limite.

Risultati Avanzati

Stone's Theorem

Teorema (Stone): Una sequenza di processi di Markov $\{X_n\}$ con generatori $\{L_n\}$ converge a processo X con generatore L se:

1. $L_n f \to Lf$ per f in core denso
2. $L_n - \lambda I$ è suriettivo per $\lambda > 0$ sufficientemente grande
3. Stime uniformi sui risolventi

Teorema di Trotter-Kato

Per semigruppi $\{T_n(t)\}$ associati a $\{X_n\}$:

$T_n(t)f \to T(t)f$ uniformemente su compatti

implica convergenza debole $X_n \Rightarrow X$.

Backward Equations e Problemi di First Passage Time

Equazioni Backward di Kolmogorov

Derivazione Probabilistica

L'equazione backward governa funzioni del valore atteso condizionate sullo stato iniziale.

Per processo di Markov $\{X(t)\}$ con generatore infinitesimale A, consideriamo:

$u(x,t) = E[g(X(T)) \mid X(t) = x]$

dove $g: S \to R$ è payoff terminale e $T > t$.

Usando proprietà di Markov:

$u(x,t) = E[E[g(X(T)) \mid X(t+h) = X(t+h)] \mid X(t) = x] = E[u(X(t+h), t+h) \mid X(t) = x]$

Per h piccolo:

$u(x,t) \approx u(x,t+h) + E[u(X(t+h), t+h) - u(x,t+h) \mid X(t) = x]$

Espandendo e prendendo limite $h \to 0$:

$\partial u/\partial t + Au = 0$

con condizione terminale $u(x,T) = g(x)$.

Forma Esplicita per Diffusioni

Per processo di diffusione:

$dX(t) = \mu(X(t))dt + \sigma(X(t))dW(t)$

l'equazione backward è:

$\partial u/\partial t + \mu(x)\partial u/\partial x + (\sigma^2(x)/2)\partial^2 u/\partial x^2 = 0$

Questa è equazione parabolica backward (tempo scorre all'indietro da T a t).

Catene a Tempo Discreto

Per catena di Markov discreta, l'equazione backward è:

$u_n(i) = \Sigma_j p_{ij} u_{n+1}(j)$

con $u_N(i) = g(i)$. Ricorsivamente:

$u_n = P u_{n+1}$

dove P è matrice di transizione.

Problemi di First Passage Time

Definizione e Setup

Il **tempo di primo passaggio** (first passage time o hitting time) all'insieme $A \subset S$ è:

$\tau_A = \inf\{t \geq 0 : X(t) \in A\}$

con convenzione $\tau_A = \infty$ se l'insieme non è mai raggiunto.

Quantità di interesse:

- **Probabilità di hitting**: $u(x) = P_x(\tau_A < \infty)$
- **Tempo medio di hitting**: $m(x) = E_x[\tau_A \mid \tau_A < \infty]$
- **Distribuzione del tempo**: $F_A(x,t) = P_x(\tau_A \leq t)$

Equazioni per Probabilità di Hitting

La probabilità $u(x) = P_x(\tau_A < \infty)$ soddisfa equazione di Dirichlet:

$Au(x) = 0$ per $x \notin A \cup \partial A$ $u(x) = 1$ per $x \in A$ $u(x) = 0$ per $x \in \partial A_\infty$ (frontiera all'infinito se transiente)

Per diffusioni:

$\mu(x)u'(x) + (\sigma^2(x)/2)u''(x) = 0$ in $(a,b) \setminus A$

con condizioni al bordo appropriate.

Per catene discrete:

$\Sigma_j p_{ij} u(j) = u(i)$ per $i \notin A$

che si riscrive come:

$(P - I)u = 0$ su $S \setminus A$, $u = 1$ su A

Equazioni per Tempi Medi

Il tempo medio $m(x) = E_x[\tau_A]$ soddisfa:

$Am(x) = -1$ per $x \notin A$ $m(x) = 0$ per $x \in A$

Per diffusioni:

$\mu(x)m'(x) + (\sigma^2(x)/2)m''(x) = -1$

Per catene discrete:

$\Sigma_j p_{ij} m(j) = m(i) + 1$ per $i \notin A$ $m(i) = 0$ per $i \in A$

che si riscrive:

$(P - I)m = -1$ su $S \setminus A$

Soluzioni Esplicite per Processi Standard

Moto Browniano con Drift

Per $dX = \mu dt + \sigma dW$, tempo di hitting di livello a partendo da $x < a$:

Probabilità di hitting (prima di $-\infty$):

$u(x) = \{ 1 \text{ se } \mu \geq 0 \quad \exp(2\mu x/\sigma^2) / \exp(2\mu a/\sigma^2) \text{ se } \mu < 0 \}$

Tempo medio (se $\mu > 0$):

$m(x) = (a - x)/\mu$

Distribuzione del tempo ($\mu = 0$):

$P_x(\tau_a \leq t) = 2\Phi((a-x)/(\sigma\sqrt{t}))$

dove Φ è CDF normale standard.

Processo di Ornstein-Uhlenbeck

Per $dX = -\theta(X - \mu)dt + \sigma dW$, hitting di livello a da x:

La soluzione coinvolge funzioni paraboliche cilindriche e non ha forma chiusa semplice in generale, ma per caso simmetrico ($x = \mu$):

$E_\mu[\tau_a] = (2/\theta) \log((a-\mu)/\sigma\sqrt{(2/\theta)})$ per $a > \mu$

Processi di Nascita-Morte

Per catena di nascita-morte con tassi λ_n, μ_n, tempo medio da i a $n > i$:

$m_i = \Sigma_{k=i}^{n-1} (1/\lambda_k) \Sigma_{j=0}^k \pi_j$

dove $\pi_j \propto (\lambda_0...\lambda_{j-1})/(\mu_1...\mu_j)$ sono pesi stazionari.

Metodi di Calcolo Numerico

Finite Differences per Diffusioni

Discretizzare l'equazione Am = -1 su griglia. Per esempio, con μ = 0:

$(\sigma^2/2) \cdot (m_{i+1} - 2m_i + m_{i-1})/h^2 = -1$

che dà sistema lineare tridiagonale risolvibile efficientemente.

Metodi Iterativi per Catene

Per catene finite, risolvere (I - P)m = 1 con $m_A = 0$.

Iterazione di Gauss-Seidel:

$m_i^{(k+1)} = 1 + \Sigma_{j \neq i} p_{ij} m_j^{(k)}$

converge se matrice è M-matrix (tipicamente vero per processi ergodici).

Simulazione Monte Carlo

Stimare m(x) simulando N traiettorie da x e mediando τ_A:

$\hat{m}(x) = (1/N) \Sigma_{k=1}^N \tau_A^{(k)}$

con errore $O(1/\sqrt{N})$. Variance reduction tramite importance sampling o splitting può ridurre drasticamente la varianza per eventi rari.

Problemi di Two-Sided Exit

Setup

Consideriamo processo su intervallo (a,b) e chiediamo: quale frontiera viene raggiunta per prima?

Definiamo:

- $\tau = \min(\tau_a, \tau_b)$: tempo di uscita dall'intervallo
- $u(x) = P_x(\tau_b < \tau_a)$: probabilità di uscire da b prima di a

Equazioni

u(x) soddisfa:

Au = 0 in (a,b) u(a) = 0, u(b) = 1

Moto browniano (μ = 0, σ = 1):

$u(x) = (x - a)/(b - a)$

Moto browniano con drift ($\mu \neq 0$):

$u(x) = [\exp(2\mu x/\sigma^2) - \exp(2\mu a/\sigma^2)] / [\exp(2\mu b/\sigma^2) - \exp(2\mu a/\sigma^2)]$

Distribuzione Congiunta

La coppia $(\tau, X(\tau))$ ha densità congiunta che può essere calcolata risolvendo sistema di PDE:

$\partial f/\partial t + Af = 0$ in $(a,b) \times (0,t)$

con condizioni al bordo riflettenti la uscita.

Problemi di Stopping Ottimo

Formulazione Generale

Trovare tempo di arresto τ che massimizza:

$V(x) = \sup_\tau E_x[e^{-r\tau} g(X(\tau))]$

dove $r > 0$ è tasso di sconto e g payoff.

Free Boundary Problems

Il valore ottimale V soddisfa:

$\max\{AV - rV, g - V\} = 0$

con **regione di continuazione** $C = \{x : V(x) > g(x)\}$ e **regione di stopping** $S = \{x : V(x) = g(x)\}$.

La frontiera ∂C tra le due regioni è free boundary da determinare come parte della soluzione.

Esempio: American Options

Per opzione put americana con strike K su asset $S(t)$:

$g(S) = (K - S)^+$ (payoff)

Processo: $dS = rSdt + \sigma SdW$ (risk-neutral)

Il prezzo $V(S,t)$ soddisfa:

$\partial V/\partial t + (rS)\partial V/\partial S + (\sigma^2 S^2/2)\partial^2 V/\partial S^2 - rV \leq 0$ $V(S,t) \geq (K - S)^+$

con uguaglianza in almeno una delle due.

Esiste **free boundary** $S^*(t)$ tale che è ottimale esercitare se $S \leq S^*(t)$.

Metodi di Approssimazione

Problema di Dirichlet Discreto

Per griglia $x_0 = a < x_1 < ... < x_n = b$, approssimare $Au = 0$:

$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = 0$

dove coefficienti vengono da discretizzazione del generatore.

Metodo delle Caratteristiche

Per equazioni backward, il metodo delle caratteristiche segue traiettorie all'indietro nel tempo, risolvendo ODE:

$dx/dt = -\mu(x)$

Trasformate di Laplace

La trasformata di Laplace del tempo di hitting:

$\varphi_\lambda(x) = E_x[e^{-\lambda\tau_A}]$

soddisfa:

$(A - \lambda)\varphi_\lambda = 0$ in $S \setminus A$ $\varphi_\lambda = 1$ su A

Invertendo la trasformata, si ottiene la distribuzione di τ_A.

Applicazioni

Rovina del Giocatore

Capitale $X(t)$ di giocatore che vince/perde unità con probabilità $p/(1-p)$. Rovina se $X = 0$ o vittoria se $X = N$.

Probabilità di rovina partendo da x:

$u(x) = \{ [(1-p)/p]^x - [(1-p)/p]^N] / [1 - [(1-p)/p]^N]$ se $p \neq 1/2$ $1 - x/N$ se $p = 1/2$ }

Tempo medio di rovina (se $p < 1/2$):

$m(x) = [x/(1-2p)] - [N/(1-2p)] \cdot [1 - [(1-p)/p]^x] / [1 - [(1-p)/p]^N]$

Credit Risk

Modello di Merton per default: asset value $V(t)$ segue diffusione, default se V tocca soglia D.

Probabilità di default entro T:

$P(\tau_D \leq T \mid V(0) = V_0) = \Phi((\log(D/V_0) + (\mu - \sigma^2/2)T)/(\sigma\sqrt{T})) + (D/V_0)^{2\mu/\sigma^2} \Phi((\log(D/V_0) - (\mu - \sigma^2/2)T)/(\sigma\sqrt{T}))$

Optimal Liquidation

Trader vuole liquidare posizione minimizzando costi. Tempo ottimale dipende da prezzo corrente e costi di transazione, modellabile come free boundary problem.

Teoria Spettrale Avanzata dei Processi di Markov

Teoria degli Operatori e Semigruppi

Un processo di Markov a tempo continuo definisce naturalmente un semigruppo di operatori lineari $\{P_t, t \geq 0\}$ che agiscono su spazi funzionali. Per funzioni misurabili f, l'operatore P_t è definito da:

$$(P_t f)(x) = E[f(X_t) \mid X_0 = x] = \int f(y)\, p(x,t;y,0)\, dy$$

dove $p(x,t;y,s)$ è il kernel di transizione. La proprietà di semigruppo $P_{t+s} = P_t P_s$ riflette direttamente la proprietà di Markov.

Il generatore infinitesimale A del semigruppo è definito come:

$$Af = \lim_{t\to 0} (P_t f - f)/t$$

quando questo limite esiste. Per processi di diffusione con drift $\mu(x)$ e coefficiente di diffusione $\sigma(x)$, il generatore è l'operatore differenziale:

$$Af(x) = \mu(x)f'(x) + (1/2)\sigma^2(x)f''(x)$$

La relazione fondamentale che collega il semigruppo al generatore è:

$$dP_t/dt = AP_t = P_t A$$

con soluzione formale $P_t = e^{tA}$. Questa equazione è analoga all'equazione di Schrödinger in meccanica quantistica, con A che gioca il ruolo dell'Hamiltoniano.

Decomposizione Spettrale e Autovalori

Per catene di Markov su spazi finiti, la matrice di transizione P può essere diagonalizzata (quando è diagonalizzabile). Se P ha autovalori $\lambda_1, \lambda_2, ..., \lambda_n$ con autovettori destri $v_1, ..., v_n$ e autovettori sinistri $u_1, ..., u_n$, allora:

$$P^n = \Sigma_{i=1}^n \lambda_i^n v_i u_i^T$$

Per catene ergodiche, $\lambda_1 = 1$ con autovettore destro $v_1 = (1,1,...,1)^T$ e autovettore sinistro $u_1 = \pi$ (distribuzione stazionaria). Tutti gli altri autovalori soddisfano $|\lambda_i| < 1$, garantendo convergenza esponenziale:

$P^n - 1\pi^T = \Sigma_{i=2}^n \lambda_i^n v_i u_i^T \to 0$

Il secondo autovalore λ_2 determina il tasso di convergenza. Per catene reversibili (soddisfacenti $\pi_i p_{ij} = \pi_j p_{ji}$), tutti gli autovalori sono reali e la matrice è simmetrica dopo riscalamento.

Teoria del Potenziale per Processi di Markov

La teoria del potenziale studia funzioni armoniche e superarmoniche rispetto al generatore di Markov. Una funzione h è armonica per A se $Ah = 0$, e superarmonica se $Ah \leq 0$.

Il potenziale di Green è definito come:

$G(x,y) = \int_0^\infty p(x,t;y,0)\, dt$

rappresentante il tempo totale atteso speso in y partendo da x (per processi transienti). Per processi ricorrenti, G è infinito ovunque. Il potenziale soddisfa l'equazione risolvente:

$(\lambda I - A)R_\lambda = I$

dove $R_\lambda(x,y) = \int_0^\infty e^{-\lambda t} p(x,t;y,0)\, dt$ è l'operatore risolvente.

La funzione di Green per il moto browniano in R^d è:

$G(x,y) = \{ (1/(2\pi)) \log|x-y| \text{ per } d=2 \quad (1/(d(d-2)\omega_d)) |x-y|^{2-d} \text{ per } d\geq 3 \quad \infty \text{ per } d=1 \}$

dove ω_d è il volume della sfera unitaria in R^d. Questo mostra che il moto browniano è ricorrente in dimensioni 1 e 2, ma transiente per $d \geq 3$.

Rappresentazione Spettrale per Processi Reversibili

Per processi di Markov reversibili rispetto a una misura μ, il generatore A è auto-aggiunto nello spazio $L^2(\mu)$. Esiste una decomposizione spettrale completa:

$Af = \Sigma_n \lambda_n \langle f, \varphi_n \rangle \varphi_n$

dove $\{\varphi_n\}$ sono autofunzioni ortonormali e $\{\lambda_n\}$ sono autovalori (tutti non positivi per generatori). Il semigruppo ha rappresentazione:

$(P_t f)(x) = \Sigma_n e^{\lambda_n t} \langle f, \varphi_n \rangle \varphi_n(x)$

Questa decomposizione permette di analizzare il comportamento temporale del processo attraverso il decadimento esponenziale dei modi. Il gap spettrale $\gamma = -\lambda_1$ (dove $\lambda_0 = 0$ corrisponde alla stazionarietà) determina il tempo di rilassamento $\tau_{rel} = 1/\gamma$.

Disuguaglianze di Poincaré e Logaritmiche di Sobolev

Per processi reversibili con distribuzione stazionaria π, la disuguaglianza di Poincaré afferma che esiste $c_P > 0$ tale che:

$\mathrm{Var}_\pi(f) \le c_P\, E(f,f)$

dove $E(f,f) = -(1/2)\int f(x)\, Af(x)\, \pi(dx)$ è la forma di Dirichlet (energetica). La costante ottimale $c_P = 1/\gamma$ è l'inverso del gap spettrale.

La disuguaglianza logaritmica di Sobolev è più forte e implica convergenza rapida in entropia relativa. Esiste $c_{LS} > 0$ tale che:

$\mathrm{Ent}_\pi(f^2) \le 2c_{LS}\, E(f,f)$

dove $\mathrm{Ent}_\pi(g) = \int g \log(g/\int g\, d\pi)\, d\pi$ è l'entropia relativa. Questa disuguaglianza implica concentrazione della misura e bound sulla tail distribution.

Per il moto browniano su $\mathbb{R}^d$ con misura gaussiana, $c_{LS} = 1$. Per processi su varietà Riemanniane, queste costanti sono legate alla curvatura di Ricci attraverso i teoremi di Bakry-Émery.

Funzioni di Lyapunov e Stabilità

Una funzione di Lyapunov $V: S \to [0,\infty)$ per un processo di Markov con generatore A soddisfa:

$AV(x) \le -\gamma V(x) + b\, I_K(x)$

per costanti $\gamma > 0$, $b < \infty$, e insieme compatto K. Questa condizione garantisce l'esistenza di una distribuzione stazionaria e convergenza esponenziale con tasso almeno γ.

Per il processo di Ornstein-Uhlenbeck $dX = -\theta X\, dt + \sigma\, dW$, la funzione quadratica $V(x) = x^2$ è Lyapunov con:

$AV = -2\theta x^2 + \sigma^2 = -2\theta V + \sigma^2$

garantendo convergenza esponenziale alla distribuzione stazionaria $N(0, \sigma^2/(2\theta))$ con tasso 2θ.

Criteri di Ergodicità di Harris

Per catene di Markov su spazi generali (non necessariamente finiti o numerabili), i criteri di Harris forniscono condizioni per ergodicità. Una catena è Harris ricorrente se esiste un insieme piccolo C (tale che le iterate della catena da C hanno componente assolutamente continua uniforme) e ogni punto comunica con C.

Il teorema ergodico di Harris afferma che per catene Harris ricorrenti aperiodiche:

$\lim_{n\to\infty} \|P^n(x,\cdot) - \pi(\cdot)\|_{TV} = 0$

per π-quasi ogni x, dove π è l'unica misura invariante. Questo generalizza il teorema ergodico classico a spazi di stati generali.

Metodi Variazionali e Teoria dell'Informazione per Processi di Markov

Introduzione ai Metodi Variazionali

I metodi variazionali trasformano problemi di inferenza e ottimizzazione in problemi di minimizzazione funzionale. Per processi di Markov, permettono di:

- Approssimare distribuzioni complesse
- Derivare algoritmi di inferenza efficienti
- Collegare meccanica statistica, teoria dell'informazione, e machine learning

Divergenze e Distanze tra Distribuzioni

Divergenza di Kullback-Leibler

La divergenza KL tra due distribuzioni p e q su S è:

$D_{KL}(p||q) = \Sigma_i p_i \log(p_i/q_i)$

oppure, per densità continue:

$D_{KL}(p||q) = \int p(x) \log(p(x)/q(x)) dx$

Proprietà:

- $D_{KL}(p||q) \geq 0$ con uguaglianza sse p = q (disuguaglianza di Gibbs)
- NON simmetrica: $D_{KL}(p||q) \neq D_{KL}(q||p)$
- NON soddisfa disuguaglianza triangolare (non è metrica)

Interpretazione Informazionale

$D_{KL}(p||q)$ rappresenta:

- **Sorpresa relativa**: Bits extra necessari per codificare dati da p usando codice ottimale per q
- **Inefficienza di codifica**: Perdita di compressione
- **Discriminabilità**: Quanto facilmente distinguere p da q

Divergenza KL Inversa

$D_{KL}(q||p)$ minimizza errore nella coda opposta:

- $D_{KL}(p||q)$: penalizza $q > 0$ dove $p = 0$ (zero-forcing)
- $D_{KL}(q||p)$: penalizza $q = 0$ dove $p > 0$ (zero-avoiding)

Mode-seeking vs Mean-seeking:

- Minimizzare D_KL(q||p): q è mode-seeking (cattura moda principale)
- Minimizzare D_KL(p||q): q è mean-seeking (spread su tutte le mode)

Altre Divergenze

Divergenza di Rényi (α > 0, α ≠ 1):

D_α(p||q) = (1/(α-1)) log Σ_i p_i^α q_i^{1-α}

Per α → 1, si recupera KL. α = 2 dà Hellinger distance.

Divergenza di Jensen-Shannon:

D_JS(p||q) = (1/2)[D_KL(p||m) + D_KL(q||m)]

dove m = (p+q)/2. È simmetrica e bounded: 0 ≤ D_JS ≤ log 2.

Distanza di Wasserstein:

W_p(μ,ν) = (inf_{γ∈Γ(μ,ν)} ∫∫ d(x,y)^p dγ(x,y))^{1/p}

dove Γ(μ,ν) sono couplings. Rilevante per optimal transport e GANs.

Principio Variazionale per Entropia

Formulazione Generale

L'entropia relativa può essere caratterizzata variazionalmente:

D_KL(p||q) = sup_f [E_p[f] - log E_q[e^f]]

dove sup è su tutte le funzioni misurabili f.

Dimostrazione: Usando disuguaglianza di Jensen, l'ottimo è raggiunto a f*(x) = log(p(x)/q(x)).

Applicazioni

Lower bound su partition function: Per distribuzione di Gibbs q(x) = e^{-E(x)}/Z:

log Z = sup_p [E_p[-E] + H(p)]

dove H(p) = -Σ p_i log p_i è entropia di Shannon.

Variational inference: Approssimare posteriore intractable p(x|y) con q(x):

log p(y) = D_KL(q||p(·|y)) + E_q[log p(y,x) - log q(x)] ≥ E_q[log p(y,x) - log q(x)] = ELBO(q)

Massimizzare ELBO (Evidence Lower Bound) equivale a minimizzare D_KL(q||p(·|y)).

Large Deviations e Principio Variazionale

Teoria delle Grandi Deviazioni

Per catena di Markov ergodica con π stazionaria, la distribuzione empirica:

$L_n = (1/n) \Sigma_{t=0}^{n-1} \delta_{X_t}$

soddisfa principio di grandi deviazioni:

$P(L_n \approx \mu) \approx \exp(-n I(\mu))$

dove $I(\mu)$ è **rate function** (entropia relativa):

$I(\mu) = D_{KL}(\mu||\pi)$

Contraction Principle

Se L_n soddisfa LDP con rate I, allora per funzione continua f:

$(1/n) \Sigma_{t=0}^{n-1} f(X_t)$ soddisfa LDP con rate $I_f(a) = \inf_{\mu: E_\mu[f]=a} I(\mu)$

Applicazione: Derivare LDP per quantità complesse da LDP di base.

Ensemble Equivalence

Per sistemi termodinamici, ensembles microcanonical e canonical sono equivalenti nel limite termodinamico via trasformata di Legendre-Fenchel della rate function.

Variational Inference per Processi di Markov

Mean-Field Approximation

Approssimare distribuzione congiunta $p(x_1,...,x_n)$ con prodotto:

$q(x_1,...,x_n) = \Pi_i q_i(x_i)$

Minimizzare $D_{KL}(q||p)$ dà equazioni auto-consistenti:

$\log q_i(x_i) = E_{q_{-i}}[\log p(x_1,...,x_n)] + \text{const}$

dove $E_{q_{-i}}$ è aspettazione rispetto a q_j per $j \neq i$.

Algoritmo Coordinate Ascent VI:

Inizializza q_i per ogni i

Repeat:

Per i = 1 a n:

q_i ← exp(E_{q_{-i}}[log p]) (normalizzata)

Until convergenza

Structured Variational Inference

Per processi di Markov, usare approssimazioni strutturate:

Markov approximation: q(x_0,...,x_T) = q(x_0) Π_t q(x_t|x_{t-1})

Preserva struttura Markoviana ma approssima transizioni.

Factorial HMM: Per multiple catene:

q(x_1,...,x_K) = Π_k q_k(x_k)

dove ogni q_k è catena di Markov.

Black Box Variational Inference

Quando E_{q_{-i}}[log p] è intrattabile, usare gradient-based optimization:

∇_θ ELBO = $\nabla_\theta E${q_θ}[log p(x,y) - log q_θ(x)]

Reparameterization trick: Se q_θ(x) = f_θ(ε) con ε ~ p(ε):

$\nabla_\theta E${q_θ}[h(x)] = E_{p(ε)}[∇_θ h(f_θ(ε))]

Low-variance, unbiased gradient estimator.

Score function estimator (REINFORCE):

$\nabla_\theta E${q_θ}[h(x)] = E_{q_θ}[h(x) ∇_θ log q_θ(x)]

Higher variance, ma più generale (non richiede riparametrizzazione).

Path Integral Formulation

Azione e Misura di Feynman-Kac

Per processo di diffusione dX = b(X)dt + σ(X)dW, la probabilità di traiettoria è data da path integral:

P[path] ∝ exp(-S[path])

dove S[path] è l'**azione**:

S[path] = (1/2) ∫_0^T [(dX/dt - b(X))^T σ^{-2}(X) (dX/dt - b(X))] dt

Most Probable Path

Il path più probabile tra x_0 e x_T minimizza l'azione:

$\delta S/\delta X(t) = 0$

dando equazioni di Eulero-Lagrange (Onsager-Machlup equations).

Applicazione: In fisica statistica, identificare transition paths tra metastable states.

Large Deviation Principle per Paths

La probabilità di osservare path X[0,T] soddisfa:

$P(X[0,T] \approx \varphi) \approx \exp(-S[\varphi]/\varepsilon^2)$

dove $\varepsilon^2 \propto$ intensità di noise. Per $\varepsilon \to 0$, dominano minimum action paths.

Entropia Relativa e Dinamica

Evoluzione Temporale dell'Entropia

Per processo di Markov con generatore L e distribuzione stazionaria π, l'entropia relativa:

$H(t) = D_{KL}(p_t||\pi)$

è monotona decrescente:

$dH/dt = -I(p_t) \leq 0$

dove I(p) è **Fisher information** (o **entropy production rate**):

$I(p) = \int (Lf/f)^2 f \, d\mu$ per $f = dp/d\pi$

Logarithmic Sobolev Inequality

Una catena soddisfa LSI con costante c_LS se per ogni distribuzione p:

$D_{KL}(p||\pi) \leq (c_{LS}/2) I(p)$

Implicazioni:

- Convergenza esponenziale: H(t) ≤ H(0) e^{-2t/c_LS}
- LSI ⇒ concentration of measure
- c_LS determina mixing time: t_mix ≤ c_LS log(1/ε)

Poincaré Inequality

Forma più debole: per ogni f con $E_\pi[f] = 0$:

$\text{Var}_\pi(f) \leq c_P E(f,f)$

dove $E(f,f) = -(1/2)\int f Lf \, d\pi$ è forma di Dirichlet.

La costante ottimale $c_P = 1/\gamma$ è inverso del gap spettrale.

Gerarchia: LSI ⇒ Poincaré ⇒ convergenza ergodica

Maximum Entropy Principle

Formulazione di Jaynes

Data constraints $E_\mu[f_i] = c_i$ per $i = 1,...,k$, la distribuzione che massimizza entropia:

$\mu^* = \text{argmax}_\mu \{H(\mu) : E_\mu[f_i] = c_i\}$

ha forma esponenziale:

$\mu^*(x) \propto \exp(\Sigma_i \lambda_i f_i(x))$

dove $\{\lambda_i\}$ sono moltiplicatori di Lagrange determinati da constraints.

Applicazioni

Statistical mechanics: Distribuzione di Boltzmann massimizza entropia dato energia media.

Maximum entropy Markov models: Data statistiche empiriche su transizioni, trovare catena che massimizza entropia condizionata.

Natural language processing: N-gram models come MaxEnt models con feature functions su context.

Dualità

Il problema primal (max entropia) è duale a:

$\min_\lambda \log Z(\lambda)$ dove $Z(\lambda) = \Sigma_x \exp(\Sigma_i \lambda_i f_i(x))$

Questa è log-partition function. Gradient: $\nabla_\lambda \log Z = E_{\mu_\lambda}[f]$.

Variational Formulation of Kolmogorov Equations

Forward Equation come Gradient Flow

L'equazione di Fokker-Planck è gradient flow dell'entropia relativa nella metrica di Wasserstein:

$\partial p/\partial t = \nabla \cdot (p \nabla (\delta F/\delta p))$

dove $F[p] = \int p \log p \, d\mu + \int V \, p \, d\mu$ è free energy.

Interpretazione: Sistema evolve per minimizzare free energy, bilanciando entropia (dispersione) e energia (concentrazione su bassi V).

JKO Scheme

Discretizzazione temporale (Jordan-Kinderlehrer-Otto):

$p_{k+1} = \text{argmin}_p \{W_2^2(p, p_k)/(2\tau) + F[p]\}$

dove τ è timestep. Questo è minimizzazione di free energy penalizzata da distanza da stato precedente.

Vantaggio: Unconditionally stable scheme per simulating gradient flows.

Control as Inference

Equivalenza tra Controllo Ottimo e Inferenza

Problema di controllo:

$\max_u E[\Sigma_t r(x_t, u_t)]$

può essere formulato come inferenza:

$p(x_{0:T}, u_{0:T} \mid O_{1:T}) \propto p(x_{0:T}, u_{0:T}) \Pi_t \exp(r(x_t, u_t))$

dove O_t è variabile binaria "optimal" (sempre osservata come true).

Inference problem: Inferire controlli u che spiegano ottimalità osservata.

Maximum Entropy RL

Maximum entropy reinforcement learning modifica obiettivo:

$\max_\pi E_\pi[\Sigma_t r(x_t, a_t) + \alpha H(\pi(\cdot|x_t))]$

dove H è entropia della policy.

Vantaggi:

- Exploration naturale (policy stocastica)
- Robustezza a perturbazioni
- Multiple solutions (portfolio di policy)

Algoritmo SAC (Soft Actor-Critic) implementa questo principio.

Inverse RL come Maximum Entropy

Dato traiettorie esperte, inferire reward function:

$\max_r [\log p(\tau|r) - \lambda||r||^2]$

Assumendo MaxEnt policy, la likelihood è:

$p(\tau|r) \propto \exp(\Sigma_t r(x_t, a_t)) / Z(r)$

Questo è log-linear model con partition function Z intractable.

Algoritmo: Gradient-based optimization alternando:

1. Policy evaluation (date r)
2. Reward update (date τ)

Diffusion Models e Score Matching

Score Function e Stein's Identity

Lo **score** di distribuzione p è:

$s(x) = \nabla_x \log p(x)$

Stein's identity: Per ogni funzione f:

$E_p[s(x)^T f(x) + \nabla \cdot f(x)] = 0$

Score Matching

Approssimare score s con modello s_θ minimizzando:

$J(\theta) = E_p[||s_\theta(x) - s(x)||^2]$

Problema: $s(x) = \nabla \log p$ è unknown!

Soluzione (Denoising score matching): Invece, minimizza:

$J_{DSM}(\theta) = E_{x,\tilde{x}}[||s_\theta(\tilde{x}) - \nabla \log q(\tilde{x}|x)||^2]$

dove $x \sim p$, $\tilde{x} \sim q(\cdot|x)$ è noisy version (es: $\tilde{x} = x + \varepsilon$, $\varepsilon \sim N(0,\sigma^2 I)$).

Diffusion Probabilistic Models

Forward process: Aggiungere noise gradualmente:

$q(x_t|x_{t-1}) = N(x_t; \sqrt{(1-\beta_t)}\, x_{t-1}, \beta_t I)$

Per T grande, $x_T \approx N(0,I)$.

Reverse process: Apprendere:

$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t,t), \Sigma_\theta(x_t,t))$

Train minimizzando:

$E_q[||\varepsilon - \varepsilon_\theta(x_t, t)||^2]$

dove ε è noise aggiunto, ε_θ predice il noise.

Generazione: Partire da $x_T \sim N(0,I)$, iterativamente denoise per ottenere $x_0 \sim p_{data}$.

Connection to Score Matching

Il reverse process può essere espresso via score:

$\mu_\theta(x_t,t) = x_t + \Sigma_\theta(x_t,t) \cdot s_\theta(x_t,t)$

Training diffusion model equivale a score matching su distribuzioni noisy intermedie.

Free Energy Principle

Variational Free Energy

Per modello generativo $p(x,z) = p(x|z)p(z)$ e posteriore approssimata $q(z|x)$:

$F(x,q) = D_{KL}(q(z|x)||p(z|x)) + \log p(x) = E_q[\log q(z|x) - \log p(x,z)]$

Minimizzare F rispetto a q dà variational inference. F è **variational free energy**.

Active Inference

Agenti cercano di minimizzare **expected free energy**:

$G = E_q[\log q(o,s) - \log p(o,s|\pi)]$

dove o sono osservazioni, s stati, π policy.

Interpretazione: Balance tra:

- Epistemic value (ridurre incertezza: $H[s|o]$)
- Pragmatic value (raggiungere goal: $E[\log p(o)]$)

Predictive Coding

Il cervello minimizza prediction error ad ogni livello gerarchico:

$\varepsilon_l = x_l - f_l(x_{l+1})$

Backpropagating errors aggiorna rappresentazioni interne, implementando variational inference neurale.

Information Geometry

Fisher Information Metric

Su manifold di distribuzioni {p_θ}, la **Fisher information matrix** è:

g_{ij}(θ) = E_p_θ[∂_i log p_θ · ∂_j log p_θ]

Definisce metrica Riemanniana: element di lunghezza infinitesimale è:

ds² = Σ_{ij} g_{ij} dθ_i dθ_j

Natural Gradient

Il **natural gradient** è:

$\tilde{\nabla}$_θ L = g^{-1}(θ) ∇_θ L

Proprietà: Invariante a reparameterization, converge più velocemente di gradient standard.

Per KL divergence: ∇_θ D_KL(p||q_θ) è natural gradient.

Exponential Families

Distribuzioni esponenziali:

p_θ(x) = exp(θ^T T(x) - A(θ))

hanno geometria particolarmente semplice:

- Fisher metric: g_{ij} = ∂_i ∂_j A(θ)
- Dualità: θ (natural) ↔ η = ∇A(θ) (expectation parameters)
- Projection: Minimizzare D_KL(p||q_θ) è proiezione euclidea in coordinate η

Applications to Machine Learning

Variational Autoencoders (VAE)

Modello:

- Encoder: q_φ(z|x) approssima posteriore
- Decoder: p_θ(x|z) genera dati

Obiettivo: Massimizzare ELBO:

$L(\theta,\varphi) = E_{q_\varphi}[\log p_\theta(x|z)] - D_{KL}(q_\varphi(z|x)||p(z))$

Training via stochastic gradient ascent con reparameterization trick.

Normalizing Flows

Trasformazioni invertibili $z = f_\theta(x)$ con:

$p_\theta(x) = p(z) |\det \partial f_\theta/\partial x|^{-1}$

Training: Massimizzare $\log p_\theta(x) = \log p(f_\theta(x)) - \log|\det J_f|$.

Architetture: RealNVP, Glow, Neural Spline Flows con Jacobians tractable.

Energy-Based Models

Distribuzioni definite da energia:

$p_\theta(x) = \exp(-E_\theta(x))/Z_\theta$

Challenge: $Z_\theta = \int \exp(-E_\theta)$ intractable.

Solutions:

- Contrastive divergence: Approssimare gradient con short MCMC chains
- Score matching: Bypass Z via matching dello score
- Noise contrastive estimation: Confrontare con distribution nota

Computational Methods

Variational Message Passing

Per grafi con struttura, propagare messaggi variazionali:

$m_{i\to j}(x_j) \propto \exp(E_{q_{-j}}[\log \psi_i(x_i,...,x_j)])$

Converge a approssimazione mean-field strutturata.

Expectation Propagation

Alternativa a variational inference che minimizza $D_{KL}(p||q)$ invece di $D_{KL}(q||p)$:

Refine iterativamente fattori di q approssimando projection:

$q^{new} \propto proj[p \cdot q^{old}/f_i] \cdot f_i^{new}$

Vantaggio: Cattura meglio code di p, ma meno stabile.

Stein Variational Gradient Descent

Particle-based variational inference: Update particles {x_i} via:

$x_i \leftarrow x_i + \varepsilon \varphi^*(x_i)$

dove $\varphi^* = \text{argmax}_\varphi \{-d/d\varepsilon D_{KL}(p_\varepsilon||\pi)|_{\varepsilon=0}\}$ subject to $||\varphi||_H \leq 1$.

Soluzione: $\varphi^*(x) = E_{x'\sim p}[k(x',x)\nabla \log \pi(x') + \nabla_{x'}k(x',x)]$

con k kernel in RKHS.

Advanced Topics

Optimal Transport e Wasserstein Gradient Flows

L'equazione di Fokker-Planck è Wasserstein gradient flow:

$\partial p/\partial t = -\nabla_W F[p]$

dove ∇_W è gradient in metrica di Wasserstein W_2.

JKO scheme: $p_{n+1} = \text{argmin}_p \{W_2^2(p,p_n)/(2\tau) + F[p]\}$

Schrödinger Bridge Problem

Trovare processo che interpola tra p_0 e p_T minimizzando:

$\int_0^T D_{KL}(P_t||R_t)\, dt$

dove R_t è reference process (es: Brownian motion).

Soluzione: Entropic optimal transport, collegato a diffusion models.

Hamilton-Jacobi-Bellman e Dualità

HJB equation per controllo ottimo:

$-\partial V/\partial t = \max_u \{L(x,u) + V_x f(x,u) + (1/2)V_{xx}\sigma^2\}$

è dual a Fokker-Planck per distribuzione ottimale.

Conclusioni

I metodi variazionali unificano:

- **Fisica statistica**: Free energy minimization, entropia massima
- **Teoria dell'informazione**: Divergenze, codifica ottimale
- **Machine learning**: VAEs, normalizing flows, score-based models
- **Controllo ottimo**: HJB equations, policy optimization
- **Inferenza bayesiana**: Variational inference, expectation propagation

Questa unificazione permette di trasferire intuizioni e tecniche tra campi diversi, creando algoritmi più potenti e generale comprensione teorica.

Metodi Variazionali e Ottimizzazione

Formulazione Variazionale dell'Equazione di Kolmogorov

L'equazione di Kolmogorov backward può essere derivata da un principio variazionale. Per un processo con generatore A e funzione terminale g, la funzione valore:

$u(x,t) = E[g(X_T) \mid X_t = x]$

minimizza il funzionale:

$J[v] = \int_t^T \int |\partial v/\partial s - Av|^2 \, d\mu \, ds + \int |v(x,T) - g(x)|^2 \, d\mu$

tra tutte le funzioni sufficientemente regolari. Questa formulazione è utile per metodi numerici basati su elementi finiti.

Entropia Relativa e Disuguaglianze Informazionali

L'entropia relativa (divergenza di Kullback-Leibler) tra due distribuzioni p e q è:

$D(p||q) = \int p(x) \log(p(x)/q(x)) \, dx$

Per processi di Markov, l'evoluzione temporale dell'entropia relativa rispetto alla distribuzione stazionaria π soddisfa:

$dD(p_t||\pi)/dt = -I(p_t)$

dove $I(p) = \int (Af/f)^2 \, f \, d\mu$ è l'informazione di Fisher (per $f = dp/d\pi$). Questo mostra che l'entropia relativa decresce monotonicamente, fornendo un'altra dimostrazione di convergenza alla stazionarietà.

Ineguaglianze di Brascamp-Lieb e Concentrazione

Per processi log-concavi (dove -log π è convessa), valgono disuguaglianze di concentrazione forti. Se $X \sim \pi$ con -log π λ-convessa, allora per funzioni Lipschitz f:

$P(|f(X) - E[f]| \geq t) \leq 2 \exp(-\lambda t^2/(2L^2))$

dove L è la costante di Lipschitz di f. Questa concentrazione esponenziale è fondamentale per analisi di algoritmi di approssimazione e sampling.

Processi Semi-Markoviani e Generalizzazioni

Processi Semi-Markov

Un processo semi-Markov generalizza processi di Markov a tempo continuo permettendo tempi di permanenza non esponenziali. Il processo è definito da:

- Catena di Markov inclusa {Y_n} con matrice di transizione P
- Tempi di permanenza {T_n} dove T_n ha distribuzione F_{Y_n, Y_{n+1}} dipendente dalla transizione

Il processo X(t) = Y_n per S_n ≤ t < S_{n+1}, dove S_n = Σ_{k=1}^n T_k sono i tempi di salto. Il kernel semi-Markov è:

Q_{ij}(t) = P(Y_{n+1} = j, T_n ≤ t | Y_n = i) = p_{ij} F_{ij}(t)

Le equazioni di rinnovo semi-Markoviane governano l'evoluzione:

p_{ij}(t) = δ_{ij}(1 - Σ_k Q_{ik}(t)) + Σ_k ∫0^t p{kj}(t-s) dQ_{ik}(s)

I processi semi-Markov sono usati per modellare affidabilità con riparazioni non immediate, code con servizi generali, e sistemi biologici con fasi di attività variabile.

Catene di Markov di Ordine Superiore

Una catena di Markov di ordine m ha:

P(X_{n+1} = j | X_n = i_n, ..., X_0 = i_0) = P(X_{n+1} = j | X_n = i_n, ..., X_{n-m+1} = i_{n-m+1})

Queste catene hanno memoria che si estende per m passi. Possono essere ridotte a catene di primo ordine espandendo lo spazio degli stati: definire Y_n = (X_n, X_{n-1}, ..., X_{n-m+1}), che è Markoviano di primo ordine.

Le catene di ordine superiore sono usate in linguistica (modelli n-gram), bioinformatica (modelli di DNA con dipendenze), e finanza (modelli autoregressive).

Processi Parzialmente Osservabili (POMDP)

Un POMDP estende MDP quando lo stato non è direttamente osservabile. Il modello è (S, A, O, T, R, Z) dove:

- S: stati (non osservabili)
- A: azioni
- O: osservazioni
- T: probabilità di transizione P(s'|s,a)
- R: funzione di ricompensa
- Z: probabilità di osservazione P(o|s,a)

La belief state b_t(s) = P(X_t = s | o_1, ..., o_t, a_1, ..., a_{t-1}) rappresenta la distribuzione di probabilità sullo stato dato le osservazioni. Il belief state evolve deterministicamente:

b_{t+1}(s') = [Σ_s T(s'|s,a_t) b_t(s)] Z(o_{t+1}|s',a_t) / P(o_{t+1}|b_t,a_t)

Risolvere POMDP è PSPACE-complete in generale. Algoritmi approssimati includono point-based value iteration e policy gradient methods.

Applicazioni Approfondite in Linguistica e Natural Language Processing

Modelli N-Gram e Language Modeling

I modelli n-gram rappresentano il linguaggio come catena di Markov dove la probabilità della prossima parola dipende dalle precedenti n-1 parole:

P(w_i | w_1, ..., w_{i-1}) ≈ P(w_i | w_{i-n+1}, ..., w_{i-1})

Per un bigram (n=2):

P(w_1, ..., w_m) = Π_{i=1}^m P(w_i | w_{i-1})

dove P(w_i|w_{i-1}) sono stimati da conteggi in corpus:

P(w_i|w_{i-1}) = Count(w_{i-1}, w_i) / Count(w_{i-1})

Tecniche di smoothing (Laplace, Good-Turing, Kneser-Ney) gestiscono bigrammi non osservati. Il perplexity misura la qualità:

Perplexity = exp(-1/N Σ log P(w_i | w_{i-1}))

Valori più bassi indicano modelli migliori.

Part-of-Speech Tagging con HMM

Il POS tagging assegna categorie grammaticali (noun, verb, adjective) a parole. Un HMM modella:

- Stati nascosti: tag POS
- Osservazioni: parole

Il modello definisce:

- Probabilità di transizione: P(tag_i | tag_{i-1})
- Probabilità di emissione: P(word_i | tag_i)

L'algoritmo di Viterbi trova la sequenza di tag ottimale. Per la frase "The dog runs":

States: {DET, NOUN, VERB} Transition: P(NOUN|DET) alta, P(VERB|NOUN) alta Emission: P("dog"|NOUN) alta, P("runs"|VERB) alta

La sequenza ottimale è DET-NOUN-VERB.

Topic Modeling con Latent Dirichlet Allocation

LDA modella documenti come miscele di topic, dove ogni topic è una distribuzione su parole. Anche se non direttamente Markoviano, l'inferenza usa Gibbs sampling (MCMC):

Per ogni parola w in documento d:

1. Rimuovi l'assegnazione corrente di topic z
2. Ricampiona z dalla distribuzione:

$$P(z_i = k \mid z_{-i}, w) \propto (n_{k,w_i}^{-i} + \beta)\,(n_{d,k}^{-i} + \alpha) / (n_k^{-i} + W\beta)$$

dove $n_{k,w}$ conta le assegnazioni di parola w a topic k, $n_{d,k}$ conta assegnazioni nel documento d a topic k.

Neural Language Models e Catene di Markov

Modelli neurali come LSTM e Transformer apprendono rappresentazioni che catturano dipendenze a lungo raggio oltre l'assunzione Markoviana. Tuttavia, la generazione autoregressive:

$$P(w_1, ..., w_n) = \Pi_{i=1}^{n} P_\theta(w_i \mid w_1, ..., w_{i-1})$$

ha struttura simile a catene di ordine infinito. La teoria di processi ergodici informa l'analisi di convergenza e proprietà statistiche di testo generato.

Applicazioni in Computer Vision

Modelli di Markov Random Fields (MRF) per Segmentazione

Gli MRF modellano immagini come grafi dove pixel sono nodi e bordi connettono pixel vicini. Per segmentazione, ogni pixel ha label (classe) e il modello definisce:

$$P(x) \propto \exp(-E(x)) = \exp(-\Sigma_i V_i(x_i) - \Sigma_{i,j} V_{ij}(x_i, x_j))$$

dove V_i sono potenziali unari (dipendenti da osservazioni) e V_{ij} sono potenziali pairwise (penalizzano discontinuità).

L'inferenza trova la configurazione ottimale $x^* = \text{argmax } P(x|obs)$. Algoritmi includono:

- Graph cuts: per potenziali submodulari
- Belief propagation: messaggi iterativi
- Mean field approximation: ottimizzazione variazionale

Hidden Markov Models per Video Analysis

I video sono naturalmente sequenze temporali. HMM modellano:

- Stati: attività/azioni (camminare, correre, saltare)
- Osservazioni: features visive (HOG, CNN features)

Per action recognition, si stima:

P(activity | video) = P(activity | o_1, ..., o_T)

usando algoritmo forward-backward. Hierarchical HMM catturano struttura a livelli multipli (movimenti → azioni → attività).

Conditional Random Fields (CRF) per Labeling

I CRF sono modelli discriminativi che condizionano direttamente su osservazioni:

P(y|x) ∝ exp(Σ_i f_i(y_i, x) + Σ_{i,j} g_{ij}(y_i, y_j, x))

dove f_i sono feature functions e g_{ij} catturano dipendenze tra label. A differenza di HMM (generativi), CRF modellano P(y|x) direttamente, evitando assunzioni su P(x).

Per image labeling, x sono pixel intensities, y sono semantic labels. L'inferenza usa:

- Viterbi per sequenze (text, 1D)
- Loopy belief propagation per grids (images, 2D)

Optical Flow e Processi di Diffusione

L'optical flow stima movimenti tra frame consecutivi. L'equazione di Horn-Schunck è una PDE variazionale:

E = ∫∫ [(I_x u + I_y v + I_t)² + λ(|∇u|² + |∇v|²)] dx dy

dove (u,v) è il campo di velocità, I_x, I_y, I_t sono gradienti dell'immagine. Aggiungendo noise, questo diventa un processo di diffusione stocastico:

du = -∇E_u dt + σ dW dv = -∇E_v dt + σ dW

L'analisi usando processi di Markov informa il comportamento di convergenza e stabilità.

Applicazioni in Neuroscienze Computazionali

Spike Trains e Processi di Punto

I neuroni comunicano tramite spike (potenziali d'azione). Lo spike train è modellato come processo di punto N(t) contante il numero di spike fino al tempo t. Il processo di Poisson è il modello base:

P(N(t) = n) = (λt)^n e^{-λt} / n!

dove λ è il tasso di firing. Modelli più realistici includono:

Processo di Poisson Non-Omogeneo: λ(t) varia nel tempo, rispondendo a stimoli: λ(t) = f(Σ_i w_i s_i(t))

dove s_i(t) sono stimoli e f è funzione di firing (sigmoidale, esponenziale).

Processo di Rinnovo: Gli intervalli inter-spike {T_i} sono iid con distribuzione generale F(t). Il coefficiente di variazione CV = σ_T/μ_T caratterizza la regolarità:

- CV < 1: firing più regolare che Poisson
- CV = 1: Poisson puro
- CV > 1: firing burstysopra Poisson

Modello di Hawkes: Processo auto-eccitatorio dove spike passati aumentano la probabilità di futuri spike:

$\lambda(t) = \mu + \Sigma_{t_i < t} \alpha \exp(-\beta(t - t_i))$

modellando bursting e dipendenza temporale.

Modelli Integrate-and-Fire

Il modello integrate-and-fire descrive il potenziale di membrana V(t):

$C\, dV/dt = -g_L(V - E_L) + I(t) + \sigma\xi(t)$

dove C è capacitanza, g_L conduttanza di leak, I(t) corrente di input, e ξ(t) è rumore bianco gaussiano. Quando V raggiunge soglia V_th, il neurone spara e V resetta a V_reset.

Tra spike, questo è processo di Ornstein-Uhlenbeck. Il tempo del primo spike è problema di first passage time. La distribuzione del tempo inter-spike è:

$f_{ISI}(t) = (1/\tau_m\sqrt{(2\pi D)}) \exp(-(V_{th} - \mu_\infty)^2/(2D)) \times \exp(-t/\tau_m) / [\Phi((\mu_\infty - V_{reset})/\sqrt{D}) - \Phi((\mu_\infty - V_{th})/\sqrt{D})]$

dove τ_m è costante di tempo di membrana, D è intensità di diffusione, μ_∞ è potenziale stazionario, e Φ è CDF normale.

Reti Neurali e Dinamiche Collettive

Reti di neuroni interagenti generano dinamiche complesse. Il modello mean-field approssima la rete come processo di Markov su (n_E(t), n_I(t)), numero di neuroni eccitatori e inibitori attivi:

$dn_E/dt = -n_E/\tau_E + f_E(w_{EE}\, n_E - w_{IE}\, n_I + I_E)$ $dn_I/dt = -n_I/\tau_I + f_I(w_{EI}\, n_E - w_{II}\, n_I + I_I)$

dove w_{XY} sono pesi sinaptici e f sono funzioni di attivazione. Analisi di stabilità rivela oscillazioni (gamma rhythm, alpha rhythm) e transizioni di fase (epilessia, anestesia).

Decodifica di Informazione Neurale

Data una sequenza di spike, inferire lo stimolo che li ha generati. Usando teorema di Bayes:

$P(s|spikes) \propto P(spikes|s)\, P(s)$

dove $P(spikes|s)$ è il modello di encoding (es: Poisson con $\lambda(s)$). L'algoritmo di Viterbi o filtro particellare stimano $s(t)$ dinamicamente.

Per popolazione di neuroni, il population vector decoder è:

$\hat{s}(t) = \Sigma_i\, n_i(t)\, c_i \,/\, \Sigma_i\, n_i(t)$

dove $n_i(t)$ è firing rate del neurone i e c_i è la sua "preferred direction" (massima risposta).

Applicazioni in Meteorologia e Climate Science

Modelli Stocastici del Clima

Il clima terrestre è sistema caotico con forte componente stocastica. Modelli energetici semplificati usano processi di Markov:

$dT/dt = (Q(1-\alpha)/4 - \varepsilon\sigma T^4)/C + \text{forcing} + \sigma_{climate}\, dW/dt$

dove T è temperatura globale, Q irradianza solare, α albedo, ε emissività, C capacità termica, e forcing rappresenta CO_2 e altri gas serra.

Questo è processo di Ornstein-Uhlenbeck non-lineare. L'analisi rivela:

- Punti di equilibrio multipli (stati climatici)
- Probabilità di transizioni (tipping points)
- Tempo di permanenza in ciascuno stato

Weather Forecasting e Assimilazione Dati

Le previsioni meteorologiche usano filtro di Kalman ensemble (EnKF) per assimilare osservazioni in modelli dinamici:

$x_{t+1} = M(x_t) + w_t$ (model forecast) $y_t = H(x_t) + v_t$ (observations)

dove M è modello meteorologico (equazioni Navier-Stokes), H è operatore di osservazione (relazione tra stato e misure).

EnKF mantiene ensemble di N stati $\{x_t^{(i)}, i=1,...,N\}$ e aggiorna ciascuno con:

$x_t^{(i)} = x_t^{(i,forecast)} + K_t(y_t + \varepsilon^{(i)} - H(x_t^{(i,forecast)}))$

dove K_t è guadagno di Kalman stimato empiricamente dall'ensemble e $\varepsilon^{(i)}$ è perturbazione delle osservazioni.

Modelli di El Niño e Oscillazioni Climatiche

El Niño Southern Oscillation (ENSO) è modellato come processo stocastico bistabile:

$$dT/dt = -aT + bT^3 + F(t) + \sigma\xi(t)$$

dove T è anomalia di temperatura nel Pacifico equatoriale, il termine cubico crea due stati attrattivi (El Niño e La Niña), F(t) è forcing stagionale, e ξ è rumore.

La distribuzione stazionaria è bimodale. Le transizioni tra stati seguono processi di Kramers con tasso:

$$\text{rate} \propto \exp(-\Delta U/\sigma^2)$$

dove ΔU è barriera di energia potenziale. Questo spiega l'irregolarità degli eventi ENSO (ogni 2-7 anni).

Modelli di Precipitazione e Processi di Punto

La precipitazione è naturalmente modellata come processo di punto alternato (Neyman-Scott):

- Tempeste arrivano secondo Poisson(λ_storm)
- Ogni tempesta genera celle di pioggia Poisson(μ_cells)
- Ogni cella dura tempo Exp(γ) con intensità Gamma(α,β)

Il processo aggregato produce precipitazione totale I(t):

$$E[I(t)] = \lambda_{storm}\, \mu_{cells}\, (\alpha/\beta)\, (1/\gamma)\, t \quad Var[I(t)] = \lambda_{storm}\, \mu_{cells}\, (\alpha/\beta^2)\, [2/\gamma - 1/\gamma^2(1 - e^{-\gamma t})]\, t$$

Questo riproduce proprietà statistiche osservate: intermittenza, clustering, correlazioni spaziali.

Applicazioni in Sistemi di Raccomandazione

Markov Chain-Based Recommendations

I sistemi di raccomandazione modellano comportamento utenti come catene di Markov dove stati sono item e transizioni rappresentano scelte sequenziali.

Per e-commerce, una sessione utente è sequenza: pagina_home → categoria_X → prodotto_A → prodotto_B → checkout. La matrice di transizione P è stimata da dati storici:

$$p_{ij} = \text{Count}(\text{transition } i \to j) / \text{Count}(\text{visits to } i)$$

La raccomandazione al passo corrente i è:

$$\text{recommend}(i) = \text{argmax}_j\, p_{ij}$$

Varianti sofisticate includono:

- **Personalized PageRank**: Random walk personalizzato con restart al profilo utente
- **Session-based RNN**: LSTM o GRU che apprendono transizioni complesse

- **Markov Decision Process**: Ottimizzano ricompense a lungo termine (lifetime value)

Collaborative Filtering e Matrix Factorization

Anche se non direttamente Markoviani, metodi CF usano MCMC per inferenza. Per matrice ratings R (utenti × item), il modello probabilistico è:

$R_{ui} = U_u^T V_i + \varepsilon_{ui}$

dove U_u e V_i sono fattori latenti e $\varepsilon \sim N(0,\sigma^2)$. Prior bayesiani:

$U_u \sim N(0, \lambda_U^{-1} I)$ $V_i \sim N(0, \lambda_V^{-1} I)$

Gibbs sampling campiona alternativamente da $P(U|V,R)$ e $P(V|U,R)$, costruendo catena di Markov con distribuzione stazionaria $P(U,V|R)$.

Temporal Dynamics e Modelli Evolutivi

Le preferenze utente evolvono nel tempo. Modelli temporali estendono MF:

$R_{ui}(t) = \mu + b_u(t) + b_i(t) + U_u(t)^T V_i(t)$

dove i fattori evolvono secondo:

$U_u(t) = U_u(t-1) + \eta_u(t), \eta \sim N(0, \sigma_U^2)$ $V_i(t) = V_i(t-1) + v_i(t), v \sim N(0, \sigma_V^2)$

Questo è processo di Markov multivariato. Il filtro di Kalman stima fattori dinamicamente incorporando nuovi ratings.

Contextual Bandits e Exploration-Exploitation

I sistemi di raccomandazione affrontano trade-off exploration-exploitation: raccomandare item popolari (exploitation) vs. esplorare preferenze (exploration).

Il multi-armed bandit con contesto modella questo come:

- Contesto: features utente/item x_t
- Azioni: item da raccomandare $a_t \in A$
- Ricompensa: click/rating r_t

L'algoritmo Upper Confidence Bound (UCB) bilancia exploration-exploitation:

$a_t = \text{argmax}_a [\hat{\mu}_a(x_t) + c\sqrt{(\log t / n_a)}]$

dove $\hat{\mu}_a$ è ricompensa stimata, n_a conteggio di volte scelta l'azione a, e c controlla exploration.

Thompson Sampling usa approccio bayesiano: mantiene distribuzione posteriore su ricompense $\theta_a \sim P(\theta_a|\text{history})$, campiona $\hat{\theta}_a \sim P(\theta_a|\text{history})$, e sceglie $a_t = \text{argmax}_a \hat{\theta}_a^T x_t$. Questa è catena di Markov nello spazio delle distribuzioni posteriori.

Network Effects e Cascate di Adozione

La diffusione di raccomandazioni in social network segue processi di Markov su grafi. Il modello Independent Cascade:

- Nodo attivo i tenta di attivare vicino j con probabilità p_{ij}
- Una sola opportunità per attivazione
- Il processo termina quando nessun nuovo nodo attivo

Questo è processo di Markov a tempo discreto su 2^n stati (sottoinsiemi di nodi attivi). La probabilità di influenza σ(S) da seed set S è:

σ(S) = E[|{nodi raggiunti partendo da S}|]

Il problema di influence maximization (trovare S di size k massimizzando σ(S)) è NP-hard ma approssimabile via simulazione Monte Carlo di cascate.

Applicazioni in Network Science e Grafi Random

Random Walks su Grafi e Centralità

Il random walk su grafo G = (V,E) è catena di Markov dove:

p_{ij} = 1/deg(i) se (i,j) ∈ E, altrimenti 0

dove deg(i) è il grado del nodo i. La distribuzione stazionaria per grafo connesso è:

π_i = deg(i) / (2|E|)

Il tempo di commute tra nodi i e j (tempo atteso per andare da i a j e tornare) è:

C_{ij} = 2|E| R_{eff}(i,j)

dove R_{eff}(i,j) è la resistenza effettiva tra i e j trattando il grafo come rete elettrica con resistenze unitarie.

PageRank e Centralità di Autovettore

PageRank definisce l'importanza di pagine web come distribuzione stazionaria di random walk con teleportazione:

P_r = (1-α)P + α(1/n)E

dove P è matrice di transizione (con handling di dangling nodes), α è probabilità di teleportazione (tipicamente 0.15), e E è matrice di tutti 1. Il PageRank è:

π = (1-α)(I - (1-α)P)^{-1} (1/n)1

Varianti includono:

- **Personalized PageRank**: Teleportazione a preferenze utente specifiche
- **Topic-Sensitive PageRank**: Diversi vettori per topic diversi
- **TrustRank**: Propagazione di trust da seed nodes fidati

Community Detection via Random Walks

Le comunità in network sono gruppi con molti link interni e pochi esterni. Il Walktrap algorithm usa random walk:

1. Calcola distanze basate su walk: d_w(i,j) misura dissimilarità tra distribuzioni di walk da i e j
2. Clustering gerarchico usando d_w
3. Ottimizza modularità Q

La modularità è:

$$Q = (1/2m) \Sigma_{ij} [A_{ij} - k_i k_j/(2m)] \delta(c_i, c_j)$$

dove A è matrice di adiacenza, k_i grado, m numero di edge, c_i comunità del nodo i.

Epidemic Spreading e Percolazione

La diffusione di epidemie/informazioni su network segue processi di Markov. Il modello SIS (Susceptible-Infected-Susceptible):

- Infezione: S → I con tasso β per contatto con infetto
- Recupero: I → S con tasso γ

Su network, lo stato è vettore x = (x_1,...,x_n) con $x_i \in \{S,I\}$. Le transizioni sono:

$$\text{Rate}(x_i: S \to I) = \beta \Sigma_{j: (i,j) \in E} x_j \quad \text{Rate}(x_i: I \to S) = \gamma$$

La soglia epidemica è:

$$\beta_c = \gamma / \lambda_1(A)$$

dove $\lambda_1(A)$ è il primo autovalore della matrice di adiacenza. Per $\beta > \beta_c$, l'infezione persiste; per $\beta < \beta_c$, si estingue.

Graph Neural Networks e Processi di Diffusione

Le GNN propagano informazione tramite message passing, analogo a processi di diffusione:

$$h_i^{(t+1)} = \sigma(W^{(t)} h_i^{(t)} + \Sigma_{j \in N(i)} M^{(t)}(h_i^{(t)}, h_j^{(t)}))$$

dove $h_i^{(t)}$ è rappresentazione del nodo i al layer t, N(i) sono vicini, M è funzione di messaggio.

Per M lineare, questo è processo di diffusione:

$$H^{(t+1)} = \sigma(W^{(t)} H^{(t)} + D^{-1/2} A D^{-1/2} H^{(t)})$$

La matrice normalizzata $D^{-1/2} A D^{-1/2}$ è matrice di transizione di random walk simmetrico. Le GNN apprendono come l'informazione si diffonde ottimalmente per il task.

Applicazioni in Biologia Molecolare e Bioinformatica

Protein Folding e Energy Landscapes

Il folding di proteine può essere modellato come processo di Markov su spazio conformazionale. Ogni conformazione è uno stato e le transizioni seguono dinamica molecolare o Monte Carlo.

Il Master Equation governa le probabilità:

$$dp_i/dt = \Sigma_j (k_{ji} p_j - k_{ij} p_i)$$

dove k_{ij} sono tassi di transizione determinati da barriere energetiche:

$$k_{ij} = k_0 \exp(-\Delta G_{ij}^{\ddagger} / k_B T)$$

dove $\Delta G_{ij}^{\ddagger}$ è energia di attivazione (transition state).

L'analisi spettrale della matrice dei tassi K rivela:

- Autovalore 0: distribuzione di equilibrio (Boltzmann)
- Gap spettrale: tempo di folding
- Autovettori: meccanismi di folding (pathways dominanti)

Markov State Models (MSM) per Dinamica Molecolare

Gli MSM discretizzano lo spazio conformazionale in microstati e stimano matrice di transizione da simulazioni MD:

$$T_{ij}(\tau) = P(X_{t+\tau} = j \mid X_t = i)$$

dove τ è lag time. I microstati sono definiti via clustering (k-means, k-centers) su spazio di features (distanze, angoli diedri).

Il test di Markovianità verifica che:

$$T(n\tau) \approx T^n(\tau)$$

L'analisi via decomposizione a implied timescales:

$$t_i = -\tau / \log|\lambda_i|$$

identifica processi lenti (folding, binding) separati da fluttuazioni veloci.

RNA Secondary Structure Prediction

La struttura secondaria di RNA (pairings di basi) evolve dinamicamente. Il modello energetico assegna energia E(s) a struttura s. La distribuzione di equilibrio è:

$$P(s) = \exp(-E(s)/RT) / Z$$

Il folding kinetics è processo di Markov tra strutture con tassi:

$$k_{s \to s'} = k_0 \exp(-\max(0, E(s') - E(s))/RT)$$

se s e s' differiscono per una base pair (move elementare).

L'algoritmo di Vienna RNA calcola P(s) esattamente per strutture senza pseudoknots via programmazione dinamica. Per kinetics, simulazione Monte Carlo o Gillespie campiona traiettorie.

Gene Regulatory Networks

Le reti di regolazione genica sono modellate come sistemi di reazioni chimiche stocastiche. Per gene i:

- Trascrizione: DNA_i → DNA_i + mRNA_i con tasso k_{on}
- Traduzione: mRNA_i → mRNA_i + Protein_i con tasso k_{tl}
- Degradazione: mRNA_i → ∅ con tasso γ_m, Protein_i → ∅ con tasso γ_p
- Regolazione: Protein_j + DNA_i → Complex (attivazione/repressione)

Lo stato è vettore di conteggi molecolari. Il Chemical Master Equation:

$$dP(n,t)/dt = \Sigma_\mu [a_\mu(n-s_\mu)P(n-s_\mu,t) - a_\mu(n)P(n,t)]$$

dove $a_\mu(n)$ è propensity della reazione μ e s_μ è stoichiometry vector.

L'algoritmo di Gillespie simula traiettorie esatte. Per sistemi grandi, approssimazioni (Langevin, diffusion approximation) riducono complessità.

Phylogenetics e Evoluzione Molecolare

La ricostruzione di alberi filogenetici usa modelli di evoluzione di sequenze come processi di Markov a tempo continuo. Lungo ciascun ramo di lunghezza t, la probabilità di sostituzione è:

$$P(j|i,t) = [\exp(Qt)]_{ij}$$

dove Q è matrice di tasso (GTR, HKY85, Jukes-Cantor).

La likelihood di albero T con topologia τ e lunghezze rami t è:

$$L(T|D) = \Sigma_{i_A} \pi_{i_A} \Pi_{branches} P(i_{child}|i_{parent}, t_{branch})$$

dove la somma è su possibili stati a nodi ancestrali A, e il prodotto su tutti i rami. L'algoritmo di Felsenstein calcola questo efficientemente via programmazione dinamica (pruning algorithm).

Metodi bayesiani (MrBayes, BEAST) usano MCMC per campionare alberi dalla posteriore:

$P(T|D) \propto P(D|T)\, P(T)$

con move proposals che modificano topologia e branch lengths, mantenendo detailed balance.

Modelli Epidemiologici Stocastici

Limitazioni dei Modelli Deterministici

I modelli deterministici (ODE) assumono popolazioni infinite e ignorano:

- Fluttuazioni stocastiche in popolazioni piccole
- Eventi rari che possono causare outbreaks
- Eterogeneità individuale e struttura di network
- Estinzione stocastica di malattie

I modelli stocastici basati su processi di Markov catturano questi fenomeni.

Modello SIR Stocastico di Base

Formulazione a Tempo Continuo

Stati: (S, I, R) con S + I + R = N.

Transizioni:

- (S, I, R) → (S-1, I+1, R) con tasso βSI/N (infezione)
- (S, I, R) → (S, I-1, R+1) con tasso γI (recupero)

Questo è processo di nascita-morte bivariato a tempo continuo.

Master Equation

La probabilità p(S,I,R,t) = P(S_t = S, I_t = I, R_t = R) soddisfa:

$dp/dt = \beta(S+1)(I-1)/N \cdot p(S+1,I-1,R,t) + \gamma(I+1) \cdot p(S,I+1,R-1,t) - [\beta SI/N + \gamma I] \cdot p(S,I,R,t)$

Per N grande, questa è intrattabile analiticamente.

Probabilità di Outbreak

Partendo da (S_0, 1, 0), qual è la probabilità di epidemia maggiore (vs estinzione precoce)?

Approssimazione branching process: Nella fase iniziale ($S \approx S_0$), ogni infetto genera Poisson(R_0) infezioni secondarie dove $R_0 = \beta S_0/(\gamma N)$.

Probabilità di estinzione precoce:

$p_{ext} = \{ 1 \text{ se } R_0 \leq 1 \; 1/R_0 \text{ se } R_0 > 1 \}$

Probabilità di outbreak: $p_{outbreak} = 1 - p_{ext}$.

Dimensione Finale dell'Epidemia

Sia R_∞ il numero finale di recuperati. Per N grande:

$E[R_\infty/N] \approx 1 - \exp(-R_0 R_\infty/N)$

Questa è equazione trascendentale che si risolve numericamente.

Per $R_0 > 1$, esiste soluzione non-banale $R_\infty/N > 0$.

Modelli con Struttura Spaziale

SIR su Network

Individui sono nodi in grafo $G = (V,E)$. Infezione si propaga solo lungo edges.

Stati: Vettore $\sigma \in \{S,I,R\}^n$ dove σ_i è stato dell'individuo i.

Transizioni:

- i: $S \to I$ con tasso $\beta \cdot |\{j: (i,j) \in E, \sigma_j = I\}|$ (infetto da vicini)
- i: $I \to R$ con tasso γ (recupero)

Message Passing Approximation

Per calcolare marginali $P(\sigma_i = I \mid t)$, usare message passing:

$m_{i \to j}^I(t)$ = probabilità che i infetti j prima del tempo t

Soddisfa equazioni integro-differenziali chiuse per grafi tree-like.

Pair Approximation

Tracciare dinamica di:

- [S], [I], [R]: densità di singoli stati
- [SI], [II], [SR], ...: densità di coppie

Equazioni di chiusura:

$d[SI]/dt \approx \beta[SI] + \bar{k}\,[S][SI]/[S] - ...$

dove $\bar{k}$ è grado medio. Più accurato di mean-field, meno costoso di simulazione full network.

Modelli Metapopolazionali

Setup

M patch connesse da migrazione. In ogni patch i, dinamica SIR locale.

Migrazione: Individui si muovono tra patch con tassi m_{ij}.

Modello completo: Stato = $(S_1, I_1, R_1, ..., S_M, I_M, R_M)$.

Processo di Markov ad alta dimensionalità (3M variabili).

Global vs Local Coupling

Coupling globale: Infezione può avvenire da infetti in qualsiasi patch (mixing completo).

Coupling locale: Infezione solo da infetti nella stessa patch (struttura spaziale).

Il coupling locale può causare:

- Viaggianti waves di infezione
- Persistence attraverso re-introduction
- Pattern spazio-temporali complessi

Source-Sink Dynamics

Patch "source" ($R_0 > 1$): Epidemia sostenuta localmente. Patch "sink" ($R_0 < 1$): Epidemia richiede importazioni esterne.

Connectivity tra source e sink determina global persistence.

Modelli con Eterogeneità

Structured SIR Model

Popolazione divisa in gruppi (età, località, behavior).

Matrice di contatto β_{ij}: tasso di contatti tra gruppo i e j.

Next Generation Matrix:

$K_{ij} = \beta_{ij} S_j^0 / (\gamma N_j)$

Numero di riproduzione di base: $R_0 = \rho(K)$ (raggio spettrale).

Esempio: Age-Structured Model

Gruppi: bambini (C), adulti (A), anziani (E).

Matrice β:

C A E

C [β_CC β_CA β_CE]

A [β_AC β_AA β_AE]

E [β_EC β_EA β_EE]

Tipicamente β_CC alto (scuole), β_AA moderato (lavoro), β_EE basso.

Vaccinazione ottimale dipende da struttura: vaccinare bambini può essere più efficace che anziani per ridurre R_0.

Modelli SEIR Stocastici

Aggiunta di Compartimento Esposto

Stati: S → E → I → R

- E: Esposti (infected ma not yet infectious)
- Periodo di latenza: tempo in E

Transizioni:

- S→E: tasso βSI/N
- E→I: tasso σE (σ = 1/latency period)
- I→R: tasso γI

Effetto della Latenza

Il periodo di latenza introduce delay, che può causare:

- Oscillazioni nelle incidenze
- Outbreaks periodici (measles, influenza)
- Harder to control (interventi delayed)

R_0 rimane β/γ ma dinamica è più complessa.

Vaccinazione e Strategie di Controllo

Modello con Vaccinazione

Nuovo compartimento V (vaccinati).

Vaccinazione:

- Alla nascita: frazione p vaccinata → V
- Campaign: S→V con tasso φ

Efficacia vaccinale: $\varepsilon \in [0,1]$

- V completamente protetti se $\varepsilon = 1$
- V parzialmente suscettibili se $\varepsilon < 1$

Herd Immunity Threshold

Per prevenire outbreak, servono abbastanza immuni:

p_c = 1 - 1/R_0

Se frazione vaccinata p > p_c, R_0^{eff} = R_0(1-p) < 1 e malattia non può invadere.

Esempio: Measles con R_0 ≈ 15 richiede p_c ≈ 93% coverage.

Strategie Ottimali

Target high-risk groups: Vaccinare gruppi con alto β o alto contatto.

Ring vaccination: Vaccinare contatti di casi confermati (Ebola, smallpox).

Pulse vaccination: Campagne periodiche invece di continuous (polio eradication).

Ottimizzazione richiede modelli stocastici per quantificare incertezza.

Modelli Multi-Strain

Competizione tra Strains

Due strains (A e B) competono per hosts suscettibili.

Stati: (S, I_A, I_B, R_A, R_B, R_{AB})

- R_A: immune solo ad A
- R_B: immune solo a B
- R_{AB}: immune ad entrambi

Cross-immunity: Parametro $\xi \in [0,1]$

- $\xi = 0$: no cross-immunity (indipendenti)
- $\xi = 1$: perfect cross-immunity (uno esclude l'altro)

Evolutionary Dynamics

Mutazioni: $I_A \rightarrow I_B$ con tasso μ.

Trade-off: Strain più trasmissibile (β alto) ma meno virulento (γ basso) vs viceversa.

Evolutionary stable strategy dipende da R_0, cross-immunity, e struttura di popolazione.

Modelli Stagionali

Time-Varying Transmission

$\beta(t) = \beta_0[1 + \alpha \cos(2\pi t/T)]$

dove $T = 1$ anno e α misura ampiezza stagionale.

Cause:

- Climatiche (influenza in inverno)
- Comportamentali (scuole aperte/chiuse)
- Immunologiche (vitamin D in inverno)

Epidemic Cycles

Per R_0 vicino a 1 e α moderato, si osservano:

- **Annual cycles**: Outbreak ogni anno nello stesso periodo
- **Biennial cycles**: Outbreak ogni 2 anni (measles pre-vaccinazione)
- **Chaotic dynamics**: Pattern irregolari (chickenpox in alcune regioni)

Bifurcation analysis identifica transizioni tra regimi.

Outbreak Prediction e Early Warning

Real-Time Estimation di R_t

R_t = numero di riproduzione effettivo al tempo t (cambia con interventi).

Stimare da dati di incidenza $I(t)$ usando:

$R_t = I(t) / \Sigma_{s=1}^{\infty} I(t-s)\, w_s$

dove w_s è generazione time distribution.

Bayesian inference: Aggiornare R_t in real-time con nuovi dati.

Early Warning Signals

Prima di outbreak, cercare:

- Increasing variance in incidence
- Increasing autocorrelation (critical slowing down)
- Rising R_t verso 1 da below

Modelli stocastici identificano questi signals anche con data limitati e noisy.

Data Assimilation

Particle Filters per Epidemie

Stato nascosto: (S_t, E_t, I_t, R_t) evolve secondo modello stocastico.

Osservazioni: Reported cases Y_t = h(I_t) + noise.

Sequential Monte Carlo:

1. Mantieni ensemble di N particelle {x_t^{(i)}}
2. Propaga con Gillespie: x_{t+1}^{(i)} ~ P(· | x_t^{(i)})
3. Peso con likelihood: w_{t+1}^{(i)} ∝ p(Y_{t+1} | x_{t+1}^{(i)})
4. Resample particelle secondo pesi

Output: Distribuzione posteriore p(x_t | Y_{1:t}) per forecasting.

Ensemble Kalman Filter (EnKF)

Approssimazione gaussiana per sistemi ad alta dimensione (metapopulation models).

Più veloce di particle filter ma meno accurato per non linearità forti.

Case Study: COVID-19

Modelli Compartimentali Estesi

SEIR + compartimenti aggiuntivi:

- A: Asintomatici (infectious ma non detectable)
- H: Ospedalizzati
- ICU: Intensive care
- D: Deceduti

Parameter estimation:

- β_A < β_I (asintomatici meno trasmissive)
- IFR (infection fatality rate) age-stratified
- Hospitalization rate dipende da healthcare capacity

Interventions

Non-pharmaceutical:

- Social distancing: riduce β
- Mask: riduce β
- Quarantine: $I \to$ isolato (non trasmette)
- Testing: riduce tempo da infection a isolation

Pharmaceutical:

- Vaccines: $S \to V$ con efficacia ε
- Therapeutics: riduce γ o IFR

Modeling impact richiede calibration a data e uncertainty quantification.

Network Models

Contact tracing: Grafo di contatti con tracciamento di edges.

Digital contact tracing: App-based, probabilistic edges.

Modelli agent-based simulano individuals con attributi dettagliati (età, household, workplace) su network realistici.

Direzioni di Ricerca Attuali

Machine Learning Hybrid Models

Combinare modelli meccanicistici (ODE/SDE/CTMC) con neural networks:

- Neural ODEs per learned dynamics
- Physics-informed neural networks (PINNs)
- Emulatori di simulatori costosi

Multi-Scale Modeling

Collegare scale:

- **Within-host**: Dinamica virale e immune response
- **Between-host**: Trasmissione nella popolazione
- **Global**: Movimento internazionale e pandemie

Modelli multi-scale catturano feedback tra scale.

Genomic Epidemiology

Integrare sequenziamento genomico:

- Filogenetica per tracciare transmission chains
- Identificare varianti emergenti
- Stimare tassi di mutazione e selezione

Phylodynamics combina phylogenetics con epidemiologia.

Climate Change e Malattie Vector-Borne

Modelli per dengue, malaria, Zika con:

- Temperatura-dependent transmission
- Shifting ranges di vectors
- Seasonal variability amplificata

Predire geographic spread e outbreak risk sotto climate scenarios.

Conclusioni sull'Epidemiologia Stocastica

I modelli stocastici basati su processi di Markov sono essenziali per:

- Catturare fluttuazioni in popolazioni finite
- Stimare probabilità di extinction vs outbreak
- Incorporare eterogeneità e struttura spaziale
- Quantificare incertezza in forecasts
- Ottimizzare strategie di controllo sotto constrained resources

Le sfide principali rimangono:

- Computational scalability per popolazioni grandi
- Parameter identifiability con data limitati
- Validazione di modelli complessi
- Communication di incertezza a policymakers

Teoria della Complessità Computazionale per Processi di Markov

Problemi Decidibili e Indecidibili

Molti problemi computazionali su catene di Markov hanno complessità nota:

PTIME (polinomiale):

- Calcolare P^n per catene finite
- Trovare distribuzione stazionaria (risolvere $\pi P = \pi$)
- Determinare ergodicity/aperiodicity per catene finite
- Calcolare tempi di hitting attesi

#P-complete (counting class):

- Calcolare probabilità di eventi rari
- Contare percorsi con proprietà specifiche
- Stimare mixing time esatto

PSPACE-complete:

- Pianificazione ottimale per POMDP
- Verificare proprietà temporali (model checking)

Indecidibile:

- Determinare se lim P^n esiste per catene infinite generali
- Verificare ergodicity per catene infinite con descrizione finita

Approssimazione e Hardness Results

Per molti problemi, l'approssimazione è più trattabile:

Mixing Time: Per catene reversibili su n stati, stimare ε-mixing time è possibile in tempo poly(n, log(1/ε), 1/γ) dove γ è gap spettrale.

Stationarie Distribution: Sampling approssimativo da π con distanza TV ≤ ε richiede O(τ_mix log(1/ε)) passi della catena.

#P-hard Approximation: Stimare P(X_n ∈ A) per eventi rari A è #P-hard anche per approssimazione, richiedendo tecniche specializzate (importance sampling, splitting).

Quantum Speed-ups

Algoritmi quantistici offrono speed-up per problemi di Markov:

Quantum Walks: Su grafi regolari, quantum walk trova elementi marked in tempo O(√N) vs O(N) classico.

HHL Algorithm: Risolve sistemi lineari (Ax = b, per trovare π da (I-P)π = 0) in tempo O(log N) vs O(N^2.37) classico, sotto condizioni (A ben-condizionata, query access).

Quantum Markov Chains: Operano su stati quantistici ρ, offrendo mixing potenzialmente più rapido per problemi specifici (quantum PageRank).

Processi di Markov in Teoria dei Giochi

Stochastic Games e Equilibri

Uno stochastic game estende MDP a giochi multi-agente. È quintupla (N, S, {A_i}, T, {R_i}) dove:

- N giocatori
- S stati
- A_i azioni giocatore i
- T(s'|s,a_1,...,a_N) transizioni
- R_i(s,a_1,...,a_N) payoff giocatore i

L'equilibrio di Nash in strategie stazionarie π = (π_1,...,π_N) soddisfa:

V_i^π(s) ≥ V_i^{π_i',π_{-i}}(s) per ogni s, i, π_i'

dove V_i^π è valore per giocatore i seguendo strategia profilo π.

Per zero-sum two-player games, esiste equilibrio in strategie pure/mixed trovabile via linear programming. Per giochi generali, esistenza è garantita ma calcolo è PPAD-complete.

Repeated Games e Folk Theorems

I repeated games sono naturalmente Markoviani: lo stato è storia passata (o credenze/reputazioni in giochi incompleti). Il folk theorem afferma che qualsiasi payoff individualmente razionale e fattibile può essere sostenuto come equilibrio di Nash subgame-perfect con sufficientemente alto discount factor.

Strategie trigger (coopera finché deviazione, poi punisci) sostengono cooperazione. La minaccia di punizione futura (attraverso transizioni di stato) mantiene incentivi.

Evolutionary Game Theory e Replicator Dynamics

La replicator dynamics modella evoluzione di strategie in popolazioni:

dx_i/dt = x_i [f_i(x) - φ(x)]

dove x_i è frequenza di strategia i, f_i payoff, φ payoff medio. Questo è processo di Markov deterministico. In versione stocastica (Moran process):

- Con probabilità x_i, individuo tipo i si riproduce
- Rimpiazza individuo random
- Include mutazioni con piccola probabilità

Gli stati absorbing sono popolazioni omogenee. La probabilità di fissazione di mutante in popolazione di N individui è:

ρ = (1 - e^{-Nσ}) / (1 - e^{-NNσ})

dove σ = (f_mutant - f_resident)/N è selective advantage.

Mean Field Games Markoviani

I mean field games (introdotti sopra) sono formulabili come limit di stochastic games con N→∞. Ciascun agente risolve MDP con transizioni e rewards dipendenti dalla distribuzione μ_t degli stati di tutti gli agenti:

max_u E[∫_0^T R(X_t, u_t, μ_t)dt] dX_t = b(X_t, u_t, μ_t)dt + σ(X_t)dW_t

L'equilibrio (u*, μ*) soddisfa:

- u* ottimale dato μ*
- μ* generato da popolazione usando u*

Questo è sistema accoppiato HJB + Fokker-Planck risolubile per giochi lineari-quadratici e alcuni casi non-lineari.

Processi di Markov e Teoria dell'Informazione

Capacità di Canali Markoviani

Un canale di comunicazione con memoria è modellato come processo di Markov. Lo stato S_t rappresenta la memoria del canale, input X_t, output Y_t. Le transizioni:

P(S_{t+1}, Y_t | S_t, X_t)

La capacità del canale (massimo rate di comunicazione affidabile) è:

C = lim_{n→∞} (1/n) max_{P(X^n)} I(X^n; Y^n)

dove I è mutua informazione. Per canali Markoviani, questo si semplifica usando struttura del processo.

Il feedback non aumenta capacità per canali memoryless, ma può aumentare capacità per canali con memoria Markoviana.

Entropia Rate di Processi di Markov

L'entropia rate di catena di Markov stazionaria è:

H(X) = lim_{n→∞} (1/n) H(X_1,...,X_n) = -Σ_{i,j} π_i p_{ij} log p_{ij}

Questo quantifica l'incertezza per simbolo nel lungo termine. Per processo ergodico:

H(X) = -E[log P(X_{n+1}|X_n)]

dove l'aspettazione è rispetto a distribuzione stazionaria congiunta.

Compressione e Codifica

Il teorema di Shannon per processi stazionari afferma che il rate ottimale di compressione senza perdita è l'entropia rate H(X). Per catene di Markov, schemi di codifica efficienti sfruttano dipendenze:

Arithmetic Coding: Codifica probabilistica usando P(X_{n+1}|X_n), ottenendo lunghezza media vicina a H(X).

Lempel-Ziv: Algoritmo universale che converge a H(X) senza conoscere le probabilità, sfruttando pattern ricorrenti (processi ergodici hanno pattern tipici).

Optimal Transport e Couplings

Due catene di Markov con matrici P e Q possono essere accoppiate via coupling: processo congiunto (X_t, Y_t) con marginali corrette. Il coupling ottimale minimizza tempo di accoppiamento (quando X_t = Y_t).

La distanza di Wasserstein tra distribuzioni μ e ν su grafo è:

$W(\mu,\nu) = \inf_{\pi \in \Gamma(\mu,\nu)} E_{(X,Y)\sim\pi}[d(X,Y)]$

dove $\Gamma(\mu,\nu)$ sono coupling con marginali μ,ν. Questa distanza quantifica dissimilarità geometrica ed è usata in:

- Generative models (Wasserstein GANs)
- Domain adaptation (optimal transport maps)
- Mixing time bounds (coupling method)

Coupling Methods per Processi di Markov

Il **coupling** (accoppiamento) è una tecnica fondamentale per studiare processi stocastici costruendo due (o più) processi sulla stessa spazio di probabilità in modo che le loro traiettorie si "incontrino" con alta probabilità.

Definizione Formale

Un **coupling** di due catene di Markov $\{X_n\}$ e $\{Y_n\}$ con matrici di transizione P e Q è un processo bivariato $\{(X_n, Y_n)\}$ tale che:

- La marginale di X_n è governata da P
- La marginale di Y_n è governata da Q
- Le due componenti possono essere dipendenti

Formalmente, esiste matrice di transizione congiunta R su $S \times S$ tale che:

$\Sigma_j R_{(i,k),(j,l)} = p_{ij}$ per ogni i,j,k $\Sigma_j R_{(i,k),(j,l)} = q_{kl}$ per ogni k,l

Tempo di Accoppiamento

Definizione

Il **tempo di accoppiamento** è:

$\tau = \inf\{n \geq 0 : X_n = Y_n\}$

Dopo τ, i due processi possono essere fatti coincidere: X_n = Y_n per ogni n ≥ τ.

Importanza

Il tempo di accoppiamento controlla la distanza in variazione totale:

$||P^n(x,\cdot) - P^n(y,\cdot)||_{TV} \leq P\{x,y\}(\tau > n)$

Questa disuguaglianza fondamentale collega convergenza e probabilità di accoppiamento.

Coupling Classico (Maximal Coupling)

Costruzione

Per due distribuzioni μ e ν su S, il **maximal coupling** massimizza P(X = Y):

$P(X = Y) = \Sigma_i \min(\mu_i, \nu_i)$

Algoritmo:

1. Genera U ~ Uniform(0,1)
2. Se U ≤ min(μ_i, ν_i)/max(μ_i, ν_i), allora X = Y = i
3. Altrimenti, genera X ~ μ indipendentemente da Y ~ ν

Proprietà

Il maximal coupling minimizza la distanza TV:

$||\mu - \nu||_{TV} = 1 - P(X = Y) = 1 - \Sigma_i \min(\mu_i, \nu_i)$

Grand Coupling per Catene di Markov

Costruzione di Doeblin

Per catena irriducibile e aperiodica, esiste n_0 e $\varepsilon > 0$ tali che:

$p_{ij}^{(n_0)} \geq \varepsilon \nu_j$ per ogni i,j

dove ν è una distribuzione.

Coupling construction:

1. Parti con $X_0 = x$, $Y_0 = y$
2. Ad ogni passo, con probabilità ε, entrambi saltano allo stesso stato secondo ν
3. Con probabilità 1-ε, evolvono indipendentemente secondo residual kernels

Risultato: $P_{x,y}(\tau \leq n) \geq 1 - (1-\varepsilon)^{\lfloor n/n_0 \rfloor}$

Coupling Comonotonico

Per catene su spazi ordinati (come R), il coupling comonotonico preserva l'ordine:

Se $X_0 \leq Y_0$, allora $X_n \leq Y_n$ per ogni n

Applicazione: Mostrare convergenza monotona a distribuzione stazionaria.

Coupling Time e Mixing Time

Relazione Fondamentale

Il **mixing time** è:

$t_{mix}(\varepsilon) = \min\{t : \max_x ||P^t(x,\cdot) - \pi||_{TV} \leq \varepsilon\}$

Vale la bound:

$t_{mix}(\varepsilon) \leq \lceil \log(1/\varepsilon) / \log(1/(1-p_{couple})) \rceil$

dove p_{couple} è la probabilità minima di coupling in un passo tra stati qualsiasi.

Teorema di Coupling

Teorema: Per ogni catena irriducibile e aperiodica:

$t_{mix}(\varepsilon) \leq E_{worst}[\tau] \cdot \log(1/\varepsilon)$

dove $E_{worst}[\tau] = \max_{x,y} E_{x,y}[\tau]$ è il tempo medio di coupling nel caso peggiore.

Path Coupling

Idea Principale

Invece di accoppiare tutti i possibili stati, si accoppiano solo stati "vicini" secondo una metrica ρ.

Teorema (Bubley-Dyer): Sia ρ metrica su S. Se per ogni coppia (x,y) con $\rho(x,y) = 1$, esiste coupling tale che:

$E[\rho(X_1, Y_1) \mid X_0 = x, Y_0 = y] \leq \beta$

con $\beta < 1$, allora:

$E[\rho(X_t, Y_t)] \leq \beta^t \cdot \rho(x,y)$

e il mixing time è $O(\log n)$ dove $n = \text{diam}(S,\rho)$.

Applicazioni

Colorazione di grafi: Dimostrare mixing rapido per Glauber dynamics su colorazioni.

Ising model: Bound sul mixing time a temperature alte.

Sampling di permutazioni: Analisi di random transpositions.

Coupling from the Past (CFTP)

Perfect Sampling

CFTP è algoritmo di Propp-Wilson (1996) che produce campioni **esattamente** dalla distribuzione stazionaria in tempo finito.

Idea: Far partire catene da tutti gli stati possibili al tempo -T e farle evolvere fino al tempo 0. Se tutte coalescono a un singolo stato x_0, allora $x_0 \sim \pi$ esattamente.

Algoritmo CFTP

$T \leftarrow 1$

Repeat:

Per ogni $x \in S$:

Simula X_t^x per $t = -T, -T+1, ..., 0$ usando stesse transizioni aleatorie

Se tutte le X_0^x coincidono:

Return X_0 (questo campione è esattamente distribuito secondo π)

Altrimenti:

$T \leftarrow 2T$ (raddoppia finestra temporale)

Proprietà chiave: Le transizioni devono essere **monotone** o usare **backward coupling** per efficienza.

Applicazioni

- Sampling da distribuzioni difficili (Ising, Potts models)
- Validazione di algoritmi MCMC approssimati
- Studio di fenomeni di phase coexistence

Strong Stationary Times

Definizione

Un tempo di arresto τ è **strong stationary time** (SST) se:

$P(X_\tau = j, \tau = t) = P(\tau = t) \cdot \pi_j$ per ogni j, t

Cioè, condizionatamente su $\{\tau = t\}$, X_τ è distribuito secondo π.

Relazione con Mixing

Se τ è SST, allora:

$||P^t(x,\cdot) - \pi||_{TV} \leq P_x(\tau > t)$

fornendo bound esplicito sul mixing time.

Costruzione

Top-to-random shuffle: Per mazzo di n carte:

1. Prendi carta in cima
2. Inseriscila in posizione uniforme random

SST: τ = primo tempo in cui la carta originalmente in posizione n viene mossa.

$E[\tau] = n \log n$, quindi $t_{mix} = O(n \log n)$.

Comparison of Markov Chains

Coupling di Processi Diversi

Date due catene con matrici P e Q, si può costruire coupling per confrontare velocità di convergenza.

Teorema: Se esiste coupling tale che $P_{x,y}(X_n \leq Y_n) = 1$ per $X_0 \leq Y_0$, e se Q è reversibile con distribuzione π, allora P converge almeno velocemente quanto Q:

$t_{mix}^P(\varepsilon) \leq t_{mix}^Q(\varepsilon)$

Applicazioni alla Ottimizzazione

Confrontare algoritmi MCMC diversi:

- Metropolis-Hastings vs Gibbs sampling
- Single-site vs block updates
- Parallel tempering vs standard MCMC

Coupling per Processi a Tempo Continuo

Reflection Coupling

Per due diffusioni $dX = b(X)dt + \sigma(X)dW$, $dY = b(Y)dt + \sigma(Y)d\tilde{W}$:

Reflection coupling: Scegliere $\tilde{W} = W$ fino a quando $X = Y$, poi riflettere.

Dopo coupling, usare stesso rumore: $\tilde{W} = W$ per mantenere $X = Y$.

Rate di Coupling

Per processi di Ornstein-Uhlenbeck:

$dX = -\theta X dt + \sigma dW$, $dY = -\theta Y dt + \sigma d\tilde{W}$

$E[|X_t - Y_t|] \leq |x - y| e^{-\theta t}$

Tempo atteso di coupling: $E[\tau] = O(1/\theta)$.

Esempi Dettagliati

Esempio 1: Random Walk su Ciclo

Stati: $\{0, 1, ..., n-1\}$ su ciclo. Ad ogni passo, muovi ± 1 con probabilità 1/2.

Coupling: Usa stesso flip di moneta per X e Y. Se $X_0 < Y_0$, allora distanza $|X_t - Y_t|$ è random walk su $\{0, ..., Y_0 - X_0\}$.

Tempo di coupling: $E[\tau] = O(n^2)$.

Mixing time: $t_{mix} = \Theta(n^2 \log n)$.

Esempio 2: Lazy Random Walk su Grafo

Stati: vertici di grafo $G = (V,E)$. Ad ogni passo: con prob. 1/2 resta, con prob. 1/2 muovi a vicino uniforme.

Path coupling: Definire $\rho(x,y)$ = distanza su grafo. Per edge (x,y):

$E[\rho(X_1, Y_1) \mid X_0 = x, Y_0 = y] \leq 1 - 1/(2d_{max})$

dove d_{max} è grado massimo.

Mixing time: $t_{mix} = O(d_{max} \cdot diam(G) \cdot \log n)$.

Esempio 3: Glauber Dynamics per Ising

Stati: configurazioni di spin $\sigma \in \{-1,+1\}^V$ su grafo.

Update: Scegli vertice v uniformemente, aggiorna σ_v secondo:

$P(\sigma_v = +1 \mid \sigma_{-v}) = \exp(\beta \Sigma_{w \sim v} \sigma_w) / [\exp(\beta \Sigma_{w \sim v} \sigma_w) + \exp(-\beta \Sigma_{w \sim v} \sigma_w)]$

Coupling: Aggiornare stesse posizioni con stesso rumore in X e Y.

Per β piccolo (alta temperatura), path coupling mostra t_mix = O(n log n).

Per β grande (bassa temperatura), mixing è esponenzialmente lento.

Tecniche Avanzate

Censored Coupling

Accoppiare solo se distanza scende sotto soglia, altrimenti evolvere indipendentemente.

Utile per spazi grandi dove full coupling è costoso.

Shift Coupling

Per processi con struttura prodotto, shift le componenti finché non coincidono.

Applicazioni a code networks, particle systems.

Wilson's Method per Uniform Spanning Trees

Algoritmo che genera uniform spanning tree usando coupling via loop-erased random walks.

Tempo atteso: O(n) per grafi planari, O(n^3) per grafi generali.

Frontiere Avanzate e Problemi Aperti

Catene di Markov Non-Reversibili

Mentre catene reversibili ($\pi_i p_{ij} = \pi_j p_{ji}$) hanno teoria spettrale ben sviluppata, catene non-reversibili appaiono naturalmente in:

- Algoritmi MCMC accelerati (Hamiltonian Monte Carlo)
- Processi driven (motori molecolari, pompe ioniche)
- Network traffic asimmetrico

La teoria per catene non-reversibili è meno completa. Domande aperte:

- Bound ottimali su mixing time senza reversibilità
- Algoritmi generali per ottimizzare convergenza rompendo reversibilità
- Teoria spettrale per operatori non-simmetrici

Processi di Salto di Markov su Varietà

I processi di diffusione su varietà Riemanniane sono ben studiati, ma processi di salto (analoghe di catene discrete) su varietà continue sono meno capiti. Applicazioni in:

- Sampling su spazi con curvatura (gruppi di Lie, spazi simmetrici)
- Geometria computazionale (mesh processing)
- Machine learning su manifolds

Domande chiave:

- Caratterizzazione di ergodicità via geometria
- Algoritmi di sampling efficienti preservando struttura geometrica
- Convergenza rate in termini di curvatura e topologia

Processi di Markov Infinito-Dimensionali

Processi su spazi funzionali (es: misure su R^d, curve in varietà) emergono in:

- Bayesian nonparametrics (Dirichlet processes)
- Stochastic PDE (filtering, control)
- Population genetics (Fleming-Viot processes)

La teoria richiede analisi funzionale avanzata (operatori su Hilbert spaces, weak convergence). Sfide:

- Caratterizzare distribuzioni stazionarie
- Proving convergenza ergodica
- Algoritmi computazionali efficienti (dimension-independent MCMC)

Machine Learning e Geometria dei Dati

Processi di Markov su grafi di dati (diffusion maps, spectral clustering) connettono topologia dati a probabilità:

$K_t(x,y) = \Sigma_i \lambda_i^t \varphi_i(x)\varphi_i(y)$

dove λ_i, φ_i sono autovalori/autovettori di operatore di diffusione. Le embedding diffusion:

$\Psi_t(x) = (\lambda_1^t \varphi_1(x), \lambda_2^t \varphi_2(x), ...)$

preservano geometric structure e rivelano clustering naturale.

Sviluppi recenti:

- Graph neural operators: Apprendere funzioni su grafi via processi di diffusione parametrizzati
- Neural ordinary/stochastic differential equations: Parametrizzare ODE/SDE con neural nets per modeling continuo
- Diffusion models: Invertire processo di diffusione per generazione (DALL-E, Stable Diffusion)

Epilogo: Unificazione e Prospettive

La teoria dei processi di Markov rappresenta uno dei pilastri unificanti della matematica applicata moderna. Dalla sua nascita concettuale nei lavori di Markov sull'analisi delle catene letterarie, si è evoluta in un framework potentissimo che attraversa virtualmente ogni dominio scientifico e tecnologico.

L'eleganza matematica della proprietà di assenza di memoria, pur essendo una idealizzazione, cattura aspetti essenziali di sistemi dinamici reali. Questa proprietà permette di costruire teorie profonde - teoria spettrale, teoremi ergodici, calcolo stocastico - che forniscono sia intuizione qualitativa sia predizioni quantitative precise.

L'ubiquità applicativa riflette il fatto che i processi di Markov modellano naturalmente il concetto di causalità locale: il futuro dipende dal presente, non esplicitamente dal passato. Questo appare in fisica (particelle che "dimenticano" la storia tra collisioni), biologia (mutazioni genetiche indipendenti dal percorso evolutivo), economia (decisioni basate su stato corrente del mercato), e innumerevoli altri contesti.

Le sfide contemporanee spingono la teoria in direzioni nuove:

1. **Scalabilità**: Sistemi con 10^6 - 10^12 stati (big data, reti sociali, simulazioni molecolari) richiedono algoritmi nuovi e approssimazioni intelligenti.

2. **Non-Markovianità**: Molti sistemi reali hanno memoria. Estensioni (processi semi-Markov, integro-differenziali, fractional) mantengono tractability catturando effetti di memoria.

3. **Struttura**: Sfruttare simmetrie, gerarchie, sparsità per analisi e computazione efficienti.

4. **Apprendimento**: Stimare modelli da dati, inferire struttura causale, predire comportamento - integrazione con machine learning moderno.

5. **Controllo e Ottimizzazione**: Progettare processi con proprietà desiderate - mixing rapido, convergenza garantita, robustezza.

La matematica dei processi di Markov continua a crescere, arricchita da nuove applicazioni che pongono domande fondamentali, e guidata da sviluppi teorici che illuminano connessioni profonde tra probabilità, analisi, algebra, geometria, e fisica. In un'era dove complessità e incertezza caratterizzano sistemi che studiamo - dai mercati finanziari al clima, dai network sociali ai genomi, dall'intelligenza artificiale al cervello - i processi di Markov rimangono strumento indispensabile per navigare e comprendere il mondo stocastico attorno a noi.